

RAGShaper: 通过自动数据合成激发复杂的 Agentic RAG 技能

Zhengwei Tao^{①✉*}, Bo Li^{①*}, Jialong Wu^{①✉}, Guochen Yan^①, Huanyao Zhang^①
Jiahao Xu^②, Haitao Mi^②, Wentao Zhang^{①†}

^①Peking University, ^②Tencent AI Lab

{tztzw, wentao.zhang}@pku.edu.cn, wujialongml@gmail.com

摘要

Agentic 检索增强生成 (RAG) 使大型语言模型能够自主规划并检索信息以解决复杂问题。然而, 由于缺乏反映真实检索环境噪声和复杂性的高质量训练数据, 构建鲁棒智能体的发展受到阻碍。传统的手动标注方法难以扩展, 且往往无法捕捉处理检索失败所需的动态推理策略。为弥合这一差距, 我们提出 RAGShaper, 一种新颖的数据合成框架, 旨在自动化构建 RAG 任务与鲁棒智能体轨迹。RAGShaper 引入了一个 InfoCurator, 用于构建富含对抗性干扰项的稠密信息树, 这些干扰项覆盖了感知和认知层级。此外, 我们提出了一种受限导航策略, 迫使教师智能体面对这些干扰项, 从而激发明确展示错误修正与噪声拒收的轨迹。全面实验表明, 基于我们合成语料库训练的模型显著优于现有基准, 在高噪声及复杂检索任务中表现出更优的鲁棒性。

1 引言

Agentic 检索增强生成 (Agentic RAG) 已成为自然语言处理领域的一项关键进展, 迅速从简单的检索-阅读流水线演进为具备复杂推理能力和动态工具使用能力的自主系统 (Jin et al., 2025; Asai et al., 2024; Li et al., 2025a; Team et al., 2025)。随着大模型 (LLMs) 在开放环境中的日益广泛应用, Agentic RAG 成

*Equal Contributions. ✉ Project Leads.

†Corresponding Author.

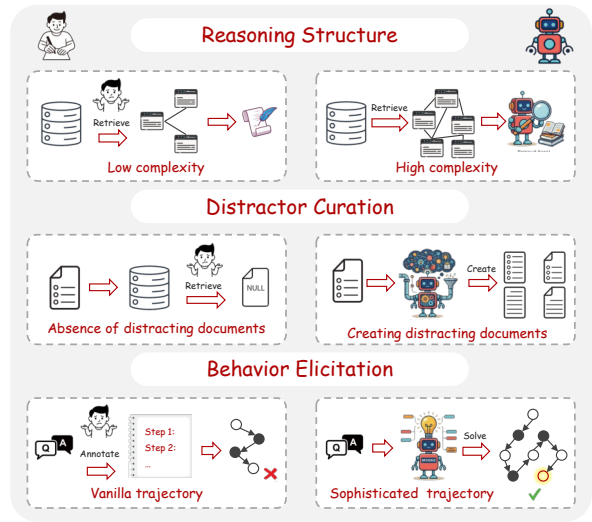


图 1: 人类对 Agentic RAG 数据集进行标注的局限性, 可通过智能体管理员的自动合成来缓解。

为了众多复杂应用的核心基础设施, 涵盖从自主研究助手到特定领域决策支持系统等多种场景。通过赋予模型主动规划检索步骤、评估获取信息并迭代优化搜索的能力, 这一范式标志着在静态知识库与智能响应之间实现跨越性突破的重要进展 (Singh et al., 2025)。

当前的方法主要依赖于人工标注的数据集, 通常以问题-轨迹-答案三元组的形式组织 (Yang et al., 2018; Ho et al., 2020)。然而, 由于人类标注者固有的认知与操作瓶颈, 这一范式从根本上不适用于训练 Agentic RAG 模型, 如图 1 所示。首先, 受限于有限的工作记忆, 标注者难以整合分散在大量异构文档中的隐含多跳证据, 往往只能采用浅层的单上下文推理, 而非实现强大智能体所需的深层检索

链 (Wu et al., 2025)。其次，手动构建真实且充满噪声的检索环境是不切实际的。那些在词汇上相似但事实错误的检索干扰项可能并不存在 (Yan et al.)。最后，人类标注难以捕捉任务分解以及从检索失败中恢复所需的动态策略调整 (Jeong et al., 2024; Tian et al., 2025)。因此，这些限制使得为 Agentic RAG 构建高质量数据在规模上难以实现。

为了克服这些障碍并实现高质量训练语料库的自动化生成，我们提出了 **RAGShaper**，这是一种专为 Agentic RAG 数据合成设计的新颖框架。针对信息构建的复杂性，RAGShaper 引入了一个 InfoCurator 模块，旨在自主构建一个全面的检索环境。从一个种子实体出发，该管理员利用检索工具在知识库中进行多轮探索，聚合出由实体及其相互关系构成的稠密信息树，以支持需要深度推理的任务合成。除了收集正面证据外，该管理员还会根据检索到的上下文动态生成对抗性的“干扰”文档。

我们系统地将这些干扰项分为两个维度，感知和认知，这一分类体系旨在培养智能体在不同层次信息噪声下的鲁棒辨别能力。完成信息整理后，大型语言模型 (LLM) 利用此结构化上下文来合成具体任务及相应的真值答案。为提取最优技能与行为模式，我们采用一个复杂的教师智能体来解决这些合成任务；尤为特别的是，我们强制实施一种受限导航策略，要求必须检索生成的干扰项，从而显式捕捉教师智能体在识别和克服信息风险方面的自适应策略。最后，通过在大规模智能体轨迹语料库上对基础模型进行微调，我们获得了一个能够在嘈杂环境中高效导航的稳健 Agentic RAG 模型。

我们总结了我们的贡献如下：

- 我们介绍了 RAGShaper，一个 Agentic RAG 数据合成框架，其包含一个 InfoCurator，旨在聚合稠密连接的信息，并在多个维度上合成复杂的检索干扰项。

- 我们提出了一种受限导航策略，以激发教师智能体的鲁棒纠错和推理行为，从而实现高质量、强韧轨迹的大规模积累。
- 我们进行了大量实验来验证我们的数据合成框架，实证结果表明，在复杂检索环境中，使用我们语料库训练的模型显著优于基准模型。

2 初步研究

我们形式化了 Agentic RAG 框架为一个自主智能体，该智能体交替进行推理与检索，能够动态地与外部语料库交互以解决知识密集型查询。采用 ReAct 范式 (Yao et al., 2023)，智能体在一系列决策过程中导航，必须迭代地弥合其内部知识与所需外部证据之间的差距。在每个时间步 t ，智能体基于初始查询和先前交互的历史，生成一个推理思考 τ_t 。这一推理指导选择特定的检索工具使用动作 α_t ，例如查询知识库 \mathbb{K} 以检索文档 \mathbb{D} ，从而获得相应的观察结果 o_t 。这一累积的推理-检索环由智能体轨迹表示，记作：

$$\mathcal{T} = (Q, \tau_1, \alpha_1, o_1, \dots, \tau_T, \alpha_T, o_T, \mathcal{A}), \quad (1)$$

其中 Q 表示用户任务，元组 (τ_i, α_i, o_i) 捕获了智能体在步骤 i 的规划、工具使用动作及反馈。 \mathcal{A} 表示针对 Q 的最终答案，代表智能体的主要目标。我们数据合成的目的是构建用于 RAG 智能体训练的 $(Q, \mathcal{A}, \mathcal{T})$ 三元组。

3 方法

我们提出 RAGShaper，一个数据合成框架，旨在自动化构建用于 Agentic RAG 的高质量训练语料库。如图 2 所示，我们的流水线包含四个阶段：(1) **信息整理** (§3.1)，其中自主的整理智能体探索初始实体以构建稠密的、含干扰项的信息树，并通过选择过程识别出有用的信息路径；(2) **问答合成** (§3.2)，从这些选定路径中衍生出任务；(3) **行为诱导** (§3.3)，教师智能体在特定干扰策略下解决这些任务，生成展现出复杂行为的轨迹；(4) **训练** (§3.4)，学生模型在这些增强后的轨迹上进行微调。

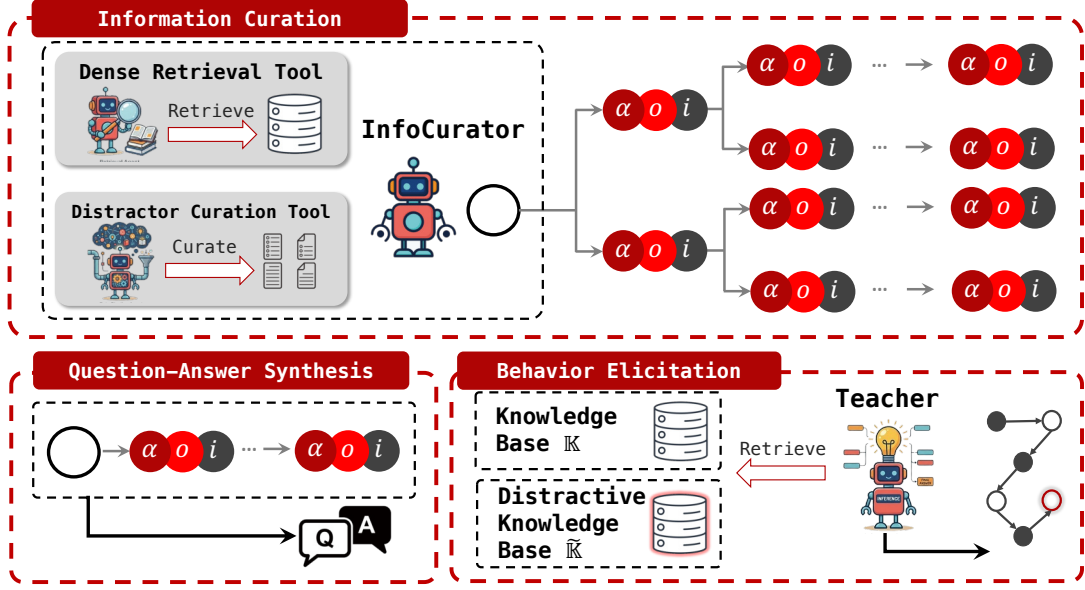


图 2: RAGShaper 概览。

3.1 信息筛选

为了训练具备深度推理能力的智能体，潜在的信息检索任务必须包含丰富的实体间关系和语义上具有挑战性的噪声。由于手动构建此类信息结构难以扩展，我们设计了一个 InfoCurator 智能体来自动化这一过程。

3.1.1 在 InfoCurator 上的树探索

InfoCurator 的目标是从知识库 \mathbb{K} 构建一个信息结构，该结构作为后续问题生成的基础。具体而言，InfoCurator 通过检索正向事实并构建干扰性文档来构建信息树。探索从一个种子实体开始，该实体作为树的根结点 s_1 。InfoCurator 随后通过深度优先遍历扩展新结点以探索新信息。一个结点定义为：

$$s_t = \begin{cases} \text{seed entity}, & t = 1 \\ \{\alpha_t, i_t, o_t\}, & t > 1, \end{cases} \quad (2)$$

其中 α_t 和 i_t 分别表示 InfoCurator 对扩展结点的动作和意图。动作 α_t 涉及检索文档或生成干扰性文档（详见下文），而 o_t 是 α_t 所导致的观察结果。我们从维基百科中爬取大规模实体¹。智能体根据从当前结点到根结点的路

径调用工具来扩展一个结点：

$$\begin{aligned} \alpha_{t+1}, i_{t+1} &= \text{InfoCurator}(\text{Path}(s_1, s_t)), \\ o_{t+1} &= \text{Execute}(\alpha_t). \end{aligned} \quad (3)$$

在每一步中，我们以概率 p^b 扩展两个子结点，以概率 $1 - p^b$ 扩展一个结点。当结点深度达到预定义的阈值时，扩展过程终止。生成的信息树包含事实及其关系。这一自动化过程显著减轻了手动数据组织的工作量。以下是 InfoCurator 使用的工具的详细说明。

稠密检索工具 InfoCurator 配备了稠密检索工具。参数包括 查询和 $Topk$ ，分别表示搜索字符串和期望的相关文档数量。该工具使用预训练的文本嵌入对查询进行编码²并计算查询与知识库中索引文档之间的相似度。它返回相似度得分超过阈值 τ 的文档，确保输出数量不超过 k ：

$$\begin{aligned} \mathbb{D} &= R(\mathbb{K}, k) \\ &= \text{Topk}(\{d \in \mathbb{K} \mid \text{sim}(\text{Query}, d) > \tau\}), \end{aligned} \quad (4)$$

其中 d 表示一个文档， τ 表示相似度阈值。

干扰项筛选工具。 一个稳健的 Agentic RAG 模型必须能够区分相关证据与噪声。仅仅在信

¹<https://www.wikipedia.org/>

²我们在 DPR 项目中使用 E5 作为检索器：
<https://github.com/facebookresearch/DPR>

Level	Type	Description	Example	Target Agent Skill
Perception Layer	Doppelgänger	Contains core topics of the query but with different metadata (version/date/ID).	Question: 2024 Financial Report. Distractor: 2025 Financial Report.	Precision Verification: Verify metadata to avoid being misled by similarity.
	False Shortcut	Forged $A \rightarrow C$ direct connection (real logic: $A \rightarrow B \rightarrow C$) with ambiguous/wrong justifications.	Truth: Virus \rightarrow Fever \rightarrow Weakness. Distractor: “Whether the virus causes weakness remains unknown ”.	Reasoning Persistence: Reject shortcuts; search for intermediate nodes.
Cognition Layer	Fragmented Puzzle	The answer is distributed across several documents.	Question: How many years has the company been profitable? Distractor: Each distractor document includes content for a single year .	Completeness Awareness: Identify information truncation; perform complete retrieval.
	Subjective Fallacy	Subjective tone with objectively wrong core arguments.	Truth: Drug X effectiveness is 95% . Distractor: Despite claims, I feel Drug X is useless.	Fact-Opinion Separation: Distinguish opinions from facts; reject unsupported views.

表 1: 干扰项类型、示例及对应的目标智能体技能。

息集中包含正面事实是不够的，我们还必须引入具有挑战性的干扰项。然而，仅依赖从知识库中检索相似事实作为干扰项往往不切实际，因为其准确率不足或缺乏合适的候选项。因此，我们提出了干扰项筛选工具，该工具可直接生成并存储具有干扰性的文档。

一个干扰文档不一定在事实上错误，但其设计目的是在 RAG 任务的上下文中造成混淆。我们包含了四种类型的干扰项，覆盖了感知和认知两个层面，如表 1 所示。该工具以原始事实、干扰类型和生成指南作为输入，调用大语言模型根据这些参数生成一个具有干扰性的事实。该指南确保生成内容的准确性。

3.1.2 信息路径选择

在构建信息结构之后，我们识别出用于问答合成的特定子结构。原始树状结构包含大量发散路径，并非所有路径都能形成连贯的推理链。我们采用启发式选择机制，从根节点到叶节点提取高价值路径。我们认为，理想的路径应具有较高的信息密度，因此我们根据每条路径所包含的文档总数（包括正向条目和干扰项）

对路径进行评分：

$$\text{score}_l = \sum_{s \in \text{Path}(s_1, s_l)} |\mathcal{D}_s|, \quad (5)$$

其中 $|\mathcal{D}_s|$ 表示结点 s 处的文档数量， s_l 为叶结点。我们选择得分最高的 m 条路径进行合成。

3.2 问题与答案的综合

路径选定后，我们合成任务 (Q, \mathcal{A}) 。为了使问题与检索步骤对齐，我们提示大语言模型“逆向工程”出一个严格依赖于路径中信息序列才能回答的问题。生成器基于完整的观测和意图序列进行条件生成：

$$(o_1^c, a_1^c, i_1, \dots, o_N^c, a_N^c, i_N) \implies (Q, \mathcal{A}) \quad (6)$$

在此，意图 i 的包含至关重要。通过显式暴露 InfoCurator 的意图，LLM 确保 Q 自然地需要特定信息，从而保证该路径作为有效的推理支持。

3.3 行为诱发

在提取出 (Q, \mathcal{A}) 对后，我们构建智能体执行轨迹 \mathcal{T} 。直接使用整理后的路径效果不佳，因为该路径可能含有噪声，或不能代表最高效

的解决方案。因此，我们采用一个 Teacher 智能体来求解 \mathcal{Q} ，从而生成最终的训练轨迹 \mathcal{T} ：

$$\tilde{\mathcal{A}}, \mathcal{T} = \text{Teacher}(\mathcal{Q}), \quad (7)$$

其中 $\tilde{\mathcal{A}}$ 为预测的答案。轨迹遵循公式 (1) 中定义的格式。教师智能体仅配备稠密检索工具，与 InfoCurator 相同。

为了激发表 1 中所描述的复杂行为和能力，我们采用了一种特定策略，利用生成的干扰项。我们将所有干扰文档聚合到一个辅助知识库 $\tilde{\mathbb{K}}$ 中。在检索过程中，该工具根据以下逻辑从原始知识库和 $\tilde{\mathbb{K}}$ 中获取文档 \mathbb{D}_t ：

$$\begin{cases} R(\mathbb{K}, k-2) \cup R(\tilde{\mathbb{K}}, 2), & \text{if } t = 1, \\ R(\mathbb{K}, k), & \text{if } \tilde{\mathbb{K}} \cap \mathbb{D}_{t-1} \neq \emptyset, \\ R(\mathbb{K}, k-2) \cup R(\tilde{\mathbb{K}}, 2), & \text{with prob. } p^e, \\ R(\mathbb{K}, k), & \text{otherwise.} \end{cases} \quad (8)$$

其中 p^e 是一个固定的概率。 $R(\mathbb{K}, k)$ 是在公式 (4) 中定义的检索函数。在第一步中，智能体被强制从 $\tilde{\mathbb{K}}$ 进行检索。如果上一步发生了从 $\tilde{\mathbb{K}}$ 的检索，则在当前步骤中被抑制，以防止连续的幻觉环。否则，从 $\tilde{\mathbb{K}}$ 进行检索的概率为 p^e 。关键的是，Teacher 智能体对 $\tilde{\mathbb{K}}$ 的存在保持未知。

3.4 训练

最后，我们将合成的三元组 $(\mathcal{Q}, \mathcal{A}, \mathcal{T})$ 编译成训练数据集，仅保留预测答案正确的轨迹（即 $\tilde{\mathcal{A}} = \mathcal{A}$ ）。我们遵循常见的 RAG 评估方法，使用 F1 得分来筛选训练数据 (Jin et al., 2025)。我们仅保留 F1 得分高于 0.9 的数据。我们对基础大语言模型进行微调，以最小化智能体轨迹 token 上的标准负对数似然损失，遵循标准的监督微调 (SFT) 协议：

$$L = -\frac{1}{\sum_{i=1}^{|\mathcal{T}|} \mathbb{I}[x_i \neq o]} \sum_{i=1}^{|\mathcal{T}|} \mathbb{I}[x_i \neq o] \cdot \log \pi_{\theta}(x_i | x_{<i}). \quad (9)$$

其中 x_i 是 i^{th} token， \mathbb{I} 是掩码观测 token 的指示函数。通过在包含自我修正和干扰项拒绝

等行为的轨迹 \mathcal{T} 上进行训练，这些行为源自我们的约束性提取过程，所得到的模型学会了在嘈杂、开放式的检索环境中自主运行。

4 实验

4.1 实验情景

数据综合。 我们将分支概率 p^b 设为 0.5，如果它位于探索树的前 2 层，否则设为 $p^b = 0$ 。树的最大深度为 30。稠密检索工具的阈值 τ 为 0.8。行为提取中的干扰概率 p^e 为 0.5。我们从每个探索树中选择两条路径 ($m = 2$) 进行数据合成。我们使用 gpt-oss-120b 作为教师智能体，其中 InfoCurator 也基于该智能体。

训练 我们在 Qwen3-30B-A3B-Think 和 Qwen3-4B-Think (Team, 2025) 上使用 Megatron-LM 框架进行训练。我们采用 4.5k 和 6k 数据集情景。详细信息见附录 A。

评估基准。 为了全面评估我们智能体的推理与检索能力，我们在四个不同的开放领域 RAG 基准上进行实验：Natural Questions (NQ) (Kwiatkowski et al., 2019)、PopQA (Mallen et al., 2023)、AmbigQA (Min et al., 2020) 和 Bamboogle (Press et al., 2023)。我们使用标准的确切的匹配 (Exact Match, EM) 和 F1 得分 (F1 Score) 指标报告性能。评估情景与 DecEx-RAG 保持一致。详细信息见附录 B。

基准。 我们对比了 RAGShaper 与大量具有竞争力的基准方法。对于基于提示的方法，我们包含了 Iter-RetGen (Shao et al., 2023)、IR-CoT (Trivedi et al., 2023)、FLARE (Jiang et al., 2023)、Selective-Context (Li, 2023)、LongLLMLingua (Jiang et al., 2024)、RECOMP (Xu et al., 2023) 和 Search-o1 (Li et al., 2025a)。针对基于学习的方法，我们与 DeepRAG (Guan et al., 2025)、IKEA (Huang et al., 2025)、ReasonRAG (Zhang et al.,

Models	Bamboogle		PopQA		NQ		AmbigQA		Avg	
	EM	F1	EM	F1	EM	F1	EM	F1	EM	F1
<i>Prompt-Based Methods</i>										
Iter-RetGen	14.4	23.9	42.5	49.3	34.5	44.2	47.0	58.8	34.6	44.1
Selective-Context	15.3	22.6	34.9	41.5	-	-	-	-	-	-
LongLLMLingua	20.3	27.4	39.2	45.1	-	-	-	-	-	-
IR-COT	16.0	27.9	32.4	39.9	19.3	35.5	24.5	40.6	23.1	36.0
RECOMP	21.7	28.6	40.5	45.8	-	-	-	-	-	-
FLARE	15.2	24.6	36.8	44.9	28.9	43.2	40.6	50.1	30.4	40.7
Search-o1	30.4	39.9	47.0	50.0	30.3	40.7	42.5	53.4	37.6	46.0
<i>Learning-Based Methods</i>										
Search-R1	30.4	43.2	41.3	46.4	36.0	45.0	49.2	60.4	39.2	48.8
IKEA	30.4	45.3	38.7	42.7	30.7	42.8	47.0	57.9	36.7	47.2
ReasonRAG	22.4	29.1	41.1	44.4	28.1	38.9	39.7	51.9	32.8	41.1
DeepRAG	-	-	40.6	43.2	-	-	-	-	-	-
DecEx-RAG	37.6	49.3	51.3	53.2	36.0	47.2	49.5	59.5	43.6	52.3
HL-Data 4.5k	50.4	67.5	35.2	48.3	31.5	47.4	52.1	69.0	42.3	58.0
<i>Ours</i>										
RAGShaper 4.5k	<u>58.5</u>	<u>70.3</u>	37.4	47.8	<u>38.3</u>	<u>50.0</u>	61.3	71.4	<u>48.8</u>	<u>59.8</u>
RAGShaper 6.5k	60.0	72.6	<u>38.9</u>	<u>49.6</u>	41.3	54.8	<u>61.1</u>	<u>71.1</u>	50.3	62.0

表 2: 在评估数据集上的性能对比。HL-Data 表示开源的人工标注数据，即从训练集中采样的 HotpotQA 和 2WikiMultiHopQA。Avg 基于 Bamboogle、PopQA、NQ 和 AmbigQA 重新计算。**加粗**表示最高得分，下划线 表示第二高得分。

2025)、DecEx-RAG (Leng et al., 2025)、Search-R1 (Jin et al., 2025) 以及 HL-Data (Jin et al., 2025; Leng et al., 2025) (即 HotPotQA 和 2Wiki 的子集, 因此不参与评估) 进行对比。详细描述见附录 C。

4.2 主要结果

RAGShaper 实现了显著的性能提升。表 2 展示了 RAGShaper 与当前最先进基准方法的对比。我们的方法始终取得最佳性能, 其中 6.5k 模型设置创造了 50.3 平均 EM 和 62.0 平均 F1 的新基准, 显著超越了基于提示 (例如, Searchol) 和基于学习的方法。

合成数据在质量上超越人工标注。关键的是, 与人工标注相比, RAGShaper 展现出更优的数据效率。在相同的数据规模 (4.5k) 下, 我们的方法在几乎所有指标上均优于 HL-Data。这表明, 我们的自动化流水线生成的训练数据质量更高, 优于传统的众包数据。

干扰项训练提升了复杂、噪声任务上的鲁棒性。性能提升在复杂且噪声密集的任务 (如 Bamboogle 和 AmbigQA) 上尤为显著。在 AmbigQA 上的显著优势直接验证了我们干扰项筛选机制和行为诱导的有效性。通过在充满多维度干扰项的轨迹上进行训练, 我们的智能体有效学会了过滤检索噪声并执行鲁棒的多跳推理, 这一能力对于应对固有的分歧并在这些挑战性数据集上动态调整检索策略至关重要。

4.3 消融研究

为了评估我们基于干扰项的学习机制的贡献, 我们进行了一项消融研究, 采用名为 RAGShaper-Dis 的变体。在数据合成阶段, 我们排除了干扰项筛选工具, 并在行为诱导阶段移除了噪声注入策略。智能体仅在干净的正向推理路径上进行训练, 未接触对抗性检索上下文。

基于干扰项的学习对于鲁棒检索至关重要。如表 3 所示, 移除这些组件会导致性能严重下降, 平均 EM 得分从 48.8 骤降至 33.8。在对噪声

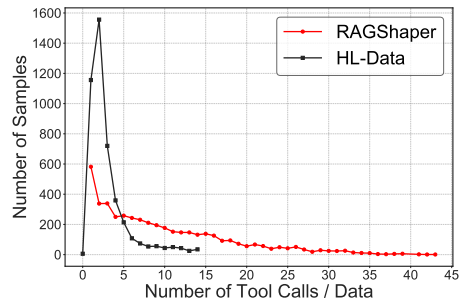


图 3: 在 RAGShaper 和 HL-Data 上对 4.5k 数据进行工具调用统计。

敏感的数据集如 AmbigQA 和 Bamboogle 上, 这种下降尤为显著。这些结果强烈凸显了我们方法的必要性: 仅在“干净”数据上进行训练不足以实现鲁棒的智能体检索。所提出的感知与认知层级干扰项的综合设计对于赋予智能体在复杂真实环境中辨别证据与噪声的关键能力至关重要。

4.4 轨迹复杂度分析

为了进一步探究我们合成语料库的推理质量, 我们分析了每条轨迹中工具使用步骤的分布情况。在图 3 中, 我们将 RAGShaper 的轨迹深度与人工标注基准 (HL-Data) 进行了对比。

RAGShaper 合成更深入、更复杂的推理任务。分布显示了任务复杂度的显著差异。HL-Data 在 2-3 步处呈现尖锐峰值, 尾部较短, 表明大多数人工标注样本代表相对浅层的 few-shot 推理任务。相比之下, RAGShaper 呈现出更宽广、长尾分布, 相当一部分轨迹需要超过 10 步, 甚至高达 40+ 步。这证实了我们的方法成功合成了更高难度的任务。

更长的轨迹编码了更丰富的智能体行为。关键的是, 更多的工具调用意味着更丰富的智能体行为密度。RAGShaper 中的长尾轨迹捕捉了复杂的认知过程, 例如导航至死胡同、验证干扰项以及进行广泛的多跳规划, 这些在简洁的 HL-Data 中很少出现。此外, 与模型可能从参数化记忆中直接作答的通用数据集不同, 我们的分布严格从零开始, 确保每条轨迹都涉及必

Models	Bamboogle		PopQA		NQ		AmbigQA		Avg	
	EM	F1	EM	F1	EM	F1	EM	F1	EM	F1
<i>Qwen3-30B-A3B-Think</i>										
RAGShaper-Dis 4.5k	38.4	58.9	27.9	42.4	28.0	44.2	41.0	61.2	33.8	51.6
RAGShaper 4.5k	58.5	70.3	37.4	47.8	38.3	50.0	61.3	71.4	48.8	59.8
<i>Qwen3-4B-Think</i>										
HL-Data 4.5k	40.8	55.3	27.0	41.8	33.5	46.8	52.9	65.6	38.5	52.4
RAGShaper 4.5k	54.4	63.9	32.7	45.4	33.1	45.0	56.0	65.5	44.0	54.9

表 3: 消融实验及在不同骨干网络上的实验。RAGShaper-Dis 指在行为诱导过程中创建并添加的干扰文档上的实验。

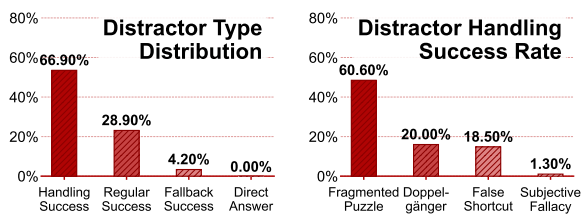


图 4: 轨迹分析。

要的检索动作。这消除了简单的“直接回答”情况，并强制执行严格的证据搜寻过程。

4.5 轨迹行为分析

为了理解我们的模型取得成功背后的机制，我们分析了合成轨迹中智能体行为的分布。我们使用大语言模型（LLM）对每条轨迹进行类型标注，结果如图 4（左）所示。

智能体依赖严格的检索而非内部知识。分析显示，大多数轨迹（66.90%）被归类为处理成功，即智能体成功识别并解决了注入的干扰项，从而得出正确答案。这一高比例结合第 4.4 节中观察到的大量工具调用，证实了我们的数据集富含高质量的 Agentic 行为。智能体并非仅进行简单的检索；而是主动对抗噪声进行推理。此外，结果表明智能体对检索存在严格依赖，而非依赖内部知识。直接回答的比例为 0.00%，而备用成功（尽管未能检索到有用信息仍正确作答）仅占 4.20%。非基于检索的答案占比极低，表明性能提升源于智能体与外部语料库交互能力的增强，而非内部知识幻觉或简单记忆。

复杂的认知陷阱为未来改进提供了空间。图 4（右）进一步剖析了由大语言模型标记的不同干扰类型的成功率，揭示出明显的难度层次。尽管智能体在解决碎片化谜题（60.60%）方面表现出色，这类任务主要测试信息整合能力，但在应对更深层次的认知陷阱时却面临显著挑战。对于虚假捷径（18.50%）和极具挑战性的主观谬误（1.30%），成功率极低，表明我们数据集的难度上限尚未达到。这一“空间”表明，RAGShaper 提供了一个足够复杂的环境，可供后续研究深入探索。未来工作可借助这些尚未被充分挖掘的复杂性，通过强化学习等先进训练范式，使智能体能够掌握这些微妙且具有对抗性的推理场景。

4.6 不同主干网络的泛化能力

为了进一步验证 RAGShaper 的有效性并不局限于特定的模型架构，我们将其评估扩展到了另一种主干网络：Qwen3-4B-Think。我们将使用我们合成数据微调的模型性能，与在相同规模（4.5k）的 HL-Data 上训练的模型进行对比。结果汇总于表 3。

RAGShaper 在多种骨干模型上展现出强大的泛化能力。如表中所示，RAGShaper 始终优于 HL-Data 基准，在整体平均得分上取得了显著提升。这证实了我们流水线生成的高质量推理轨迹具有普遍益处且可迁移，而非仅针对特定实验模型的特性进行过拟合。

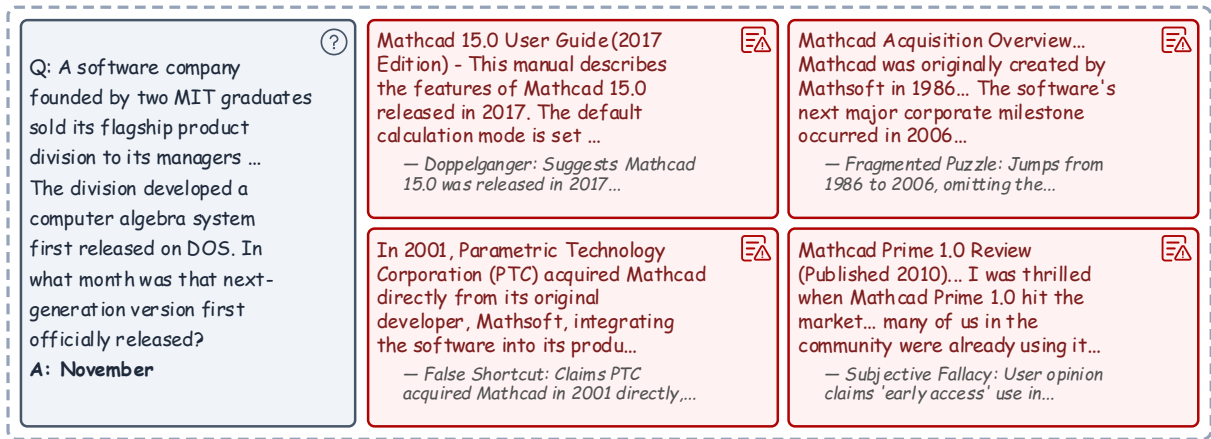


图 5: 数据合成中使用的干扰项分类的示例。该图展示了四种不同类型的认知陷阱（双重化身、碎片拼图、虚假捷径和主观谬误），旨在挑战智能体的检索与推理鲁棒性。

4.7 案例研究

图 5 展示了一个带有干扰文档的问答案例。我们添加了这些干扰文档能够引发复杂行为的原因。我们的方法能够生成多样且有效的干扰项，以激发 RAG 智能体的高级能力。

5 相关工作

检索增强推理方法 现有工作通过基于提示和基于学习的方法提升了 RAG。基于提示的方法在不更新模型参数的情况下增强推理，包括将检索与思维链推理交织 (Shao et al., 2023; Trivedi et al., 2023)、根据生成置信度自适应触发检索 (Jiang et al., 2023), 以及压缩上下文以提高信息效率 (Li, 2023; Jiang et al., 2024; Xu et al., 2023; Lee et al.). 最近, 诸如 Search-o1 等专有系统将检索工具直接集成到推理过程中, 并取得了具有竞争力的性能 (Li et al., 2025a; Sun et al.). 基于学习的方法通过训练智能体来协调检索与生成, 进一步提升性能, 通常将该过程建模为马尔可夫决策过程 (Guan et al., 2025; Huang et al., 2025), 或应用过程监督的强化学习进行细粒度最优化 (Zhang et al., 2025; Leng et al., 2025)。此外, 像 Search-R1 这样的强开源模型为推理主干配备了可训练的搜索能力 (Jin et al., 2025)。

RAG 的数据。 高质量的 RAG 系统通常依赖于人工标注的监督。标准基准 (Jin et al., 2025; Leng et al., 2025; Yu et al., 2024; Li et al., 2025b) 使用如 HotpotQA 和 2WikiMultiHopQA (Yang et al., 2018; Ho et al., 2020) 这类数据集, 这些数据集经过人工精心整理, 用于测试多跳推理能力。然而, 构建此类数据集需要大量的人工标注工作来验证证据链, 因此成本高昂且难以扩展用于通用训练。

智能体式数据合成。 为解决训练通用智能体时的数据稀缺问题, 近期研究转向了智能体数据合成, 其中利用智能体生成高质量的训练样本 (Gao et al., 2025; Chen et al., 2025; Zhai et al., 2025)。Auto-Explorer (Guo et al., 2025) 提出了一种探索者智能体, 能够自主导航并解析 GUI 环境, 无需人工干预即可收集多样化的状态-动作对。类似地, OS-Genesis (Sun et al., 2025) 提出了逆向任务合成流水线, 智能体首先与环境交互以生成轨迹, 随后将这些轨迹回溯性地与合成的高层指令对齐。对于搜索智能体, WebShaper (Tao et al., 2025) 采用形式化驱动的框架, 并结合智能体扩展器, 迭代生成复杂的查询和推理路径。此外, DeepSeek-V3.2 (Liu et al., 2025) 实现了一个大规模合成流水线, 部署专用智能体在多个领域内构建并验证任务, 从而提升智能体的泛化能力。

6 结论

我们提出了 RAGShaper, 这是一个旨在克服人类标注在 Agentic RAG 中可扩展性和质量限制的框架。通过利用 InfoCurator, 我们自动化构建了稠密检索环境, 该环境在感知和认知维度上均包含对抗性干扰项。此外, 我们的约束导航策略能够有效捕捉教师智能体的鲁棒纠错行为。实验结果表明, 在我们合成的语料库上训练的模型在复杂情景中显著优于基准模型。

局限性

在本工作中, 我们利用 RAGShaper 构建了 RAG 智能体的复杂行为。然而, 如第 4.5 节所述, 我们的数据尚未完全释放其潜力。在未来的工作中, 可以对我们的数据应用更先进的方法, 并结合进一步的训练机制。

道德考虑

本工作使用公开可用的维基百科文档和实体, 不会包含任何命名或唯一识别个人的信息, 也不会包含不当内容。我们仅将人工智能用作写作助手。

References

- Akari Asai, Zeqiu Wu, Yizhong Wang, Avirup Sil, and Hannaneh Hajishirzi. 2024. Self-rag: Learning to retrieve, generate, and critique through self-reflection.
- Xuanzhong Chen, Zile Qiao, Guoxin Chen, Liangcai Su, Zhen Zhang, Xinyu Wang, Pengjun Xie, Fei Huang, Jingren Zhou, and Yong Jiang. 2025. Agentfrontier: Expanding the capability frontier of llm agents with zpd-guided data synthesis. *arXiv preprint arXiv:2510.24695*.
- Jiaxuan Gao, Wei Fu, Minyang Xie, Shusheng Xu, Chuyi He, Zhiyu Mei, Banghua Zhu, and Yi Wu. 2025. Beyond ten turns: Unlocking long-horizon agentic search with large-scale asynchronous rl. *arXiv preprint arXiv:2508.07976*.
- Xinyan Guan, Jiali Zeng, Fandong Meng, Chunlei Xin, Yaojie Lu, Hongyu Lin, Xianpei Han, Le Sun, and Jie Zhou. 2025. Deeprag: Thinking to retrieve step by step for large language models. *arXiv preprint arXiv:2502.01142*.
- Xiangwu Guo, Difei Gao, and Mike Zheng Shou. 2025. Auto-explorer: Automated data collection for gui agent. *arXiv preprint arXiv:2511.06417*.
- Xanh Ho, Anh-Khoa Duong Nguyen, Saku Sugawara, and Akiko Aizawa. 2020. Constructing a multi-hop qa dataset for comprehensive evaluation of reasoning steps. *arXiv preprint arXiv:2011.01060*.
- Ziyang Huang, Xiaowei Yuan, Yiming Ju, Jun Zhao, and Kang Liu. 2025. Reinforced internal-external knowledge synergistic reasoning for efficient adaptive search agent. *arXiv preprint arXiv:2505.07596*.
- Soyeong Jeong, Jinheon Baek, Sukmin Cho, Sung Ju Hwang, and Jong C Park. 2024. Adaptive-rag: Learning to adapt retrieval-augmented large language models through question complexity. In *Proceedings of the 2024 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 1: Long Papers)*, pages 7029–7043.
- Huiqiang Jiang, Qianhui Wu, Xufang Luo, Dongsheng Li, Chin-Yew Lin, Yuqing Yang, and Lili Qiu. 2024. Longllmlingua: Accelerating and enhancing llms in long context scenarios via prompt compression. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1658–1677.
- Zhengbao Jiang, Frank F Xu, Luyu Gao, Zhiqing Sun, Qian Liu, Jane Dwyer, and Graham Neubig. 2023. Active retrieval augmented generation. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 7969–7992.
- Bowen Jin, Hansi Zeng, Zhenrui Yue, Jinsung Yoon, Sercan Arik, Dong Wang, Hamed Zamani, and Jiawei Han. 2025. Search-rl: Training llms to reason and leverage search engines with reinforcement learning. *arXiv preprint arXiv:2503.09516*.
- Jeff Johnson, Matthijs Douze, and Hervé Jégou. 2019. Billion-scale similarity search with GPUs. *IEEE Transactions on Big Data*, 7(3):535–547.
- Tom Kwiatkowski, Jennimaria Palomaki, Olivia Redfield, Michael Collins, Ankur Parikh, Chris Alberti, Danielle Epstein, Illia Polosukhin, Jacob Devlin, Kenton Lee, and 1 others. 2019. Natural questions: A benchmark for question answering research. *Transactions*

- of the Association for Computational Linguistics, 7:453–466.
- Meng-Chieh Lee, Qi Zhu, Costas Mavromatis, Zhen Han, Soji Adeshina, Vassilis N Ioannidis, Huzefa Rangwala, and Christos Faloutsos. Agent-g: An agentic framework for graph retrieval augmented generation.
- Yongqi Leng, Yikun Lei, Xikai Liu, Meizhi Zhong, Bojian Xiong, Yurong Zhang, Yan Gao, Yao Hu, Deyi Xiong, and 1 others. 2025. Decex-rag: Boosting agentic retrieval-augmented generation with decision and execution optimization via process supervision. In *Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing: Industry Track*, pages 1412–1425.
- Xiaoxi Li, Guanting Dong, Jiajie Jin, Yuyao Zhang, Yujia Zhou, Yutao Zhu, Peitian Zhang, and Zhicheng Dou. 2025a. Search-o1: Agentic search-enhanced large reasoning models. *arXiv preprint arXiv:2501.05366*.
- Yuan Li, Qi Luo, Xiaonan Li, Bufan Li, Qinyuan Cheng, Bo Wang, Yining Zheng, Yuxin Wang, Zhangyue Yin, and Xipeng Qiu. 2025b. R3-rag: Learning step-by-step reasoning and retrieval for llms via reinforcement learning. *arXiv preprint arXiv:2505.23794*.
- Yucheng Li. 2023. Unlocking context constraints of llms: Enhancing context efficiency of llms with self-information-based content filtering. *arXiv preprint arXiv:2304.12102*.
- Aixin Liu, Aoxue Mei, Bangcai Lin, Bing Xue, Bingxuan Wang, Bingzheng Xu, Bochao Wu, Bowei Zhang, Chaofan Lin, Chen Dong, and 1 others. 2025. Deepseek-v3. 2: Pushing the frontier of open large language models. *arXiv preprint arXiv:2512.02556*.
- Alex Mallen, Akari Asai, Victor Zhong, Rajarshi Das, Daniel Khashabi, and Hannaneh Hajishirzi. 2023. When not to trust language models: Investigating effectiveness and limitations of parametric and non-parametric memories. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 9802–9818.
- Sewon Min, Julian Michael, Hannaneh Hajishirzi, and Luke Zettlemoyer. 2020. AmbigQA: Answering ambiguous open-domain questions. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 5783–5797.
- Ofir Press, Shikhar Murty, Srinivasan Iyer, Mike Lewis, Wen-tau Yih, and Omer Levy. 2023. Measuring and narrowing the compositional gap in language models. In *Findings of the Association for Computational Linguistics: EMNLP 2023*, pages 5687–5711.
- Zhihong Shao, Yeyun Gong, Yelong Shen, Minlie Huang, Nan Duan, and Weizhu Chen. 2023. Enhancing retrieval-augmented large language models with iterative retrieval-generation synergy. In *Findings of the Association for Computational Linguistics: EMNLP 2023*, pages 9248–9274.
- Aditi Singh, Abul Ehtesham, Saket Kumar, and Tala Talaei Khoei. 2025. Agentic retrieval-augmented generation: A survey on agentic rag. *arXiv preprint arXiv:2501.09136*.
- Lei Sun, Zhengwei Tao, Youdi Li, and Hiroshi Arakawa. Oda: Observation-driven agent for integrating llms and knowledge graphs.
- Qiushi Sun, Kanzhi Cheng, Zichen Ding, Chuanyang Jin, Yian Wang, Fangzhi Xu, Zhenyu Wu, Chengyou Jia, Liheng Chen, Zhoumianze Liu, and 1 others. 2025. Osgenesis: Automating gui agent trajectory construction via reverse task synthesis. In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 5555–5579.
- Zhengwei Tao, Jialong Wu, Wenbiao Yin, Junkai Zhang, Baixuan Li, Haiyang Shen, Kuan Li, Liwen Zhang, Xinyu Wang, Yong Jiang, and 1 others. 2025. Webshaper: Agentially data synthesizing via information-seeking formalization. *arXiv preprint arXiv:2507.15061*.
- Qwen Team. 2025. [Qwen3 technical report](#). Preprint, arXiv:2505.09388.
- Tongyi DeepResearch Team, Baixuan Li, Bo Zhang, Dingchu Zhang, Fei Huang, Guangyu Li, Guoxin Chen, Huifeng Yin, Jialong Wu, Jingren Zhou, and 1 others. 2025. Tongyi deepresearch technical report. *arXiv preprint arXiv:2510.24701*.
- Fangzheng Tian, Jinyuan Fang, Debasis Ganguly, Zaiqiao Meng, and Craig Macdonald. 2025. Am i on the right track? what can predicted query performance tell us about the search behaviour of agentic rag. *arXiv preprint arXiv:2507.10411*.
- Harsh Trivedi, Niranjana Balasubramanian, Tushar Khot, and Ashish Sabharwal. 2023. Interleaving retrieval with chain-of-thought reasoning for knowledge-intensive multi-step questions. In *Proceedings of the 61st annual meeting of the association for computational linguistics (volume 1: long papers)*, pages 10014–10037.
- Jialong Wu, Baixuan Li, Runnan Fang, Wenbiao Yin, Liwen Zhang, Zhengwei Tao, Dingchu Zhang, Zekun Xi, Gang Fu, Yong Jiang, and

- 1 others. 2025. Webdancer: Towards autonomous information seeking agency. *arXiv preprint arXiv:2505.22648*.
- Fangyuan Xu, Weijia Shi, and Eunsol Choi. 2023. Recomp: Improving retrieval-augmented lms with compression and selective augmentation. *arXiv preprint arXiv:2310.04408*.
- Shi-Qi Yan, Jia-Chen Gu, Yun Zhu, and Zhen-Hua Ling. Corrective retrieval augmented generation.
- Zhilin Yang, Peng Qi, Saizheng Zhang, Yoshua Bengio, William Cohen, Ruslan Salakhutdinov, and Christopher D Manning. 2018. Hotpotqa: A dataset for diverse, explainable multi-hop question answering. In *Proceedings of the 2018 conference on empirical methods in natural language processing*, pages 2369–2380.
- Shunyu Yao, Jeffrey Zhao, Dian Yu, Nan Du, Izhak Shafran, Karthik Narasimhan, and Yuan Cao. 2023. React: Synergizing reasoning and acting in language models. In *International Conference on Learning Representations (ICLR)*.
- Yue Yu, Wei Ping, Zihan Liu, Boxin Wang, Jiaxuan You, Chao Zhang, Mohammad Shoeybi, and Bryan Catanzaro. 2024. Rankrag: Unifying context ranking with retrieval-augmented generation in llms. *Advances in Neural Information Processing Systems*, 37:121156–121184.
- Yunpeng Zhai, Shuchang Tao, Cheng Chen, Anni Zou, Ziqian Chen, Qingxu Fu, Shinji Mai, Li Yu, Jiaji Deng, Zouying Cao, and 1 others. 2025. Agentevolver: Towards efficient self-evolving agent system. *arXiv preprint arXiv:2511.10395*.
- Wenlin Zhang, Xiangyang Li, Kuicai Dong, Yichao Wang, Pengyue Jia, Xiaopeng Li, Yingyi Zhang, Derong Xu, Zhaocheng Du, Huifeng Guo, and 1 others. 2025. Process vs. outcome reward: Which is better for agentic rag reinforcement learning. *arXiv preprint arXiv:2505.14069*.

A 训练细节

我们对 Qwen3-30B-A3B-Think 进行微调。³ Qwen3-4B-Think⁴ 使用 Megatron-LM 框架的模型。我们将上下文长度扩展至 128k。我们采用 AdamW 优化器，配置为精度感知模式，并结合余弦衰减学习率调度器。该调度器具有 1.0×10^{-5} 的峰值学习率、 1.0×10^{-6} 的极小学习率，以及 5% 的预热阶段。全局批量大小配置为：Qwen3-30B-A3B-Think 为 16，Qwen3-4B-Think 为 40。两个模型均训练 5 轮次，选择表现最佳的检查点进行评估。

A.1 评价指标

按照标准的开放域问答系统协议，我们采用两种主要指标：

- **确切匹配 (EM)**: 在规范化后，衡量与一个真实答案完全匹配的预测所占的百分比。
- **F1 得分**: 衡量预测答案与真实值之间的 token 重叠程度，提供对部分正确的细致评估。

B 评估基准

我们使用四个数据集来评估检索和推理的不同方面：

- **自然问题 (NQ) (Kwiatkowski et al., 2019)**: 一个大规模基准，包含用户实际向 Google 搜索发起的查询。我们采用开放域划分，要求智能体从整个维基百科语料库中检索答案。
- **PopQA (Mallen et al., 2023)**: 旨在评估长尾实体的事实检索能力。该数据集包含参数化记忆通常无法处理的查询，因此需要进行精确的外部检索。

³<https://huggingface.co/Qwen/Qwen3-30B-A3B-Thinking-2507>

⁴<https://huggingface.co/Qwen/Qwen3-4B-Thinking-2507>

- **AmbigQA (Min et al., 2020)**: 源自 NQ，该数据集专注于存在多种合理答案的模糊查询。它挑战智能体在消除用户意图歧义以及在嘈杂的检索上下文中导航的能力。
- **Bamboogle (Press et al., 2023)**: 一个“谷歌无法破解”的数据集，旨在测试多跳推理能力。该数据集中的问题需要从多个不同的文档中综合信息，而非寻找单的直接答案。

C 基准详情

我们将其方法与以下几种具有竞争力的基准方法进行比较：

基于提示的方法。 这些方法利用固定的大型语言模型（大模型）结合先进的提示工程或检索策略：

- **Iter-RetGen (Shao et al., 2023)**: 迭代地协同检索与生成，利用模型输出来优化后续的检索查询。
- **IR-CoT (Trivedi et al., 2023)**: 将思维链推理与检索步骤交错进行，以指导多跳问答系统。
- **FLARE (Jiang et al., 2023)**: 一种主动检索策略，仅在模型生成低置信度的 token 时触发信息查询。
- **上下文最优化方法**: 包括 **选择性上下文 (Li, 2023)**、**LongLLMLingua (Jiang et al., 2024)** 和 **RECOMP (Xu et al., 2023)**，这些方法专注于压缩和选择上下文，以优化信息流至生成器。
- **Search-o1 (Li et al., 2025a)**: 一种专有的基准，采用配备搜索工具的 OpenAI o1-preview 模型，代表了推理时推理能力的最先进水平。

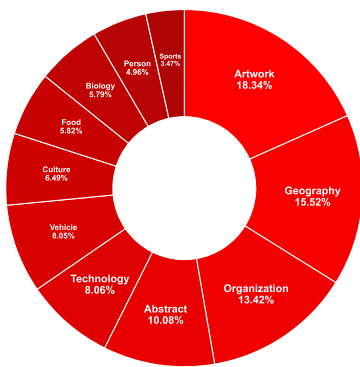


图 6: 领域分布。

基于学习的方法。 这些涉及对智能体或检索器进行训练以提升性能：

- **DeepRAG (Guan et al., 2025):** 将检索增强推理建模为马尔可夫决策过程 (MDP)，以实现自适应检索。
- **宜家 (Huang et al., 2025):** 一种强化智能体，旨在将内部参数化知识与外部搜索相结合，以优化效率。
- **ReasonRAG (Zhang et al., 2025):** 采用过程监督的强化学习，并使用细粒度奖励进行查询和答案生成。
- **DecEx-RAG (Leng et al., 2025):** 通过过程监督实现决策与执行最优化，提升 Agentic RAG 性能。
- **Search-R1 (Jin et al., 2025):** 采用具备搜索功能的 DeepSeek-R1 模型，作为强开源权重推理模型的代表。
- **HL-Data:** 一种在高质量人工标注数据集（结合 HotpotQA 和 2WikiMulti-HopQA）上微调的监督基准模型 (Yang et al., 2018; Ho et al., 2020)。该模型与我们合成数据的规模相匹配，用作数据质量的对照。

D 部署与推理细节

我们部署了 gpt-oss-120b⁵ 我们使用 vLLM 推理引擎在 8× 张 H20 GPU 上运行训练好的模型。对于 gpt-oss-120b，我们将最大上下文长度设置为 100,000 token。工具调用解析器配置为使用 openai 格式。对于我们的训练模型，我们采用了 hermes 工具调用解析器。所有模型均通过兼容 OpenAI 的 API 提供服务，以保持一致的接口。我们使用 FAISS (Johnson et al., 2019) 来支持快速相似度搜索。

E 领域多样性分析

为验证我们合成语料库的语义覆盖范围和泛化潜力，我们对生成的任务进行了领域分析。我们使用大语言模型自动对数据集中的每个问答对的主题进行分类。

RAGShaper 在领域多样性方面取得了卓越且均衡的表现。如图 6 所示，生成的分布表明 RAGShaper 具有卓越的多样性。合成任务覆盖了广泛的知识领域，从人文学科和社会科学（例如 *Artwork*、*Culture*、*Organization*）到理工科领域（例如 *Technology*、*Biology*）。值得注意的是，该分布极为均衡；没有任何单一领域主导语料库，最大类别 (*Artwork*) 仅占 18.34%，而像 *Abstract* 这样复杂的类别也具有显著的代表 (10.08%)。这种结构上的均匀性确保了训练后的智能体不会对特定实体类型产生过拟合，而是能够获得适用于多样化现实场景的泛化检索与推理能力。

⁵<https://huggingface.co/openai/gpt-oss-120b>

Trajectory Example: Handling Doppelgänger Entities

用户问题：一位在 1508 年即位的神圣罗马帝国皇帝委托创作了一部重要的文学作品，该作品构成了其宏大的艺术遗产的一部分。尽管这位赞助人在这一时期委托制作了著名的手稿文集，但这部特定的寓言史诗却特别为印刷机设计，以确保更广泛的读者群体。其首版的确切出版年份是什么？

[Thought 1] 我需要根据约束条件确定作品及其创作日期……我将查找马西米利安一世在此时期委托的作品……

[动作 1] 稠密 _ 搜索 (查询 =”马克西米利安一世委托的文学作品，完成于 1508-1519 年，手稿与印刷版对比”)

[观察 1]

候选人 A (来源 1)：

“《阿姆布拉斯英雄书》……由汉斯·里德从 1504 年到 1516 年撰写。它是一部辉煌的手稿……”

[分析：潜在的双重人格] 完成度 (1516) 符合统治时期。然而，该来源明确将其标识为“手稿”，这与印刷品约束相冲突。

候选人 B (来源 2)：

“寓意史诗《蒂尔丹克》……第一版于 1517 年在纽伦堡印刷并出版……”

[分析] 出版物 (1517 年) 符合统治时期。格式 (印刷) 符合约束条件。

[思考 2] 检索到的文献提到了马克西米利安一世在这一时期相关的两部主要作品：阿姆布拉斯英雄书和蒂尔达恩克。

我需要严格遵守格式约束。关于 *Ambraser Heldenbuch* 的段落明确将其描述为“手稿”，这与用户要求提供“出版年份”（暗示为印刷版）的要求相矛盾。

相比之下，*Theuerdank* 的描述指出它于 1517 年“印刷”并“出版”。这与所有约束一致。

[动作 2] 稠密 _ 搜索 (查询 =”Theuerdank 第一版出版年份”)

[观察 2] 来源 3: “...《特沃德安克》(1517 年)是他大规模印刷项目中的第一部...”

[思考 3] 证据确凿……1. 捐赠者：马克西米利安一世。2. 日期：1517 年。3. 格式：印刷。

[最终答案] 1517

图 7: “Doppelgänger Handling” 情景的详细轨迹。智能体成功区分了由马克西米利安一世同时委托创作的两件作品。注意：标有 [分析] 颜色块为作者添加的标注，用于说明批判性区别（手稿与印刷品），并不代表智能体的原始观察。

Trajectory Example: Countering Subjective Fallacy


用户问题：2014 年，科学界流传一份非正式提案，建议以汉弗里·戴维的名字命名一种合成元素。该元素的临时 IUPAC 名称为“ununpentium”，其发现于 2015 年 12 月得到认可，其正式名称在 2016 年 11 月的某一天获得批准，当时发现者行使了命名权。该元素的正式名称是什么？

[思考 1] 我需要确定临时名称为“ununpentium”（元素 115）的元素的正式名称。关键约束：1. 2015 年 12 月获得认可。2. 正式名称于 2016 年 11 月获批。3. 曾有以“Davy”命名的提案，但我需要确认这是否为最终获批的名称，还是仅传闻。

初步资料显示元素 115 名为“镆”，但提到“汉弗里·戴维”需加以核实，以确保自己不会陷入关于被拒绝或流行的非正式命名提议的陷阱问题。

[动作 1] 稠密 _ 搜索 (查询 = “Moscovium 名称 2016 年 11 月批准官方名称元素 115”)

[观察 1]

来源 2 & 5: “……2016 年 11 月 28 日，它被正式命名，以莫斯科州命名……” / “2016 年 6 月，国际纯粹与应用化学联合会（IUPAC）批准了该提议……并于 2016 年 11 月 28 日正式接受。”



源 3 (数据冲突)：

“化学界终于在 2016 年承认了汉弗里·戴维爵士的遗产，当时国际纯粹与应用化学联合会（IUPAC）正式批准将元素 115 命名为‘戴维姆’...”

[分析：潜在谬误] 此片段声称“达维姆”已获批准。这与来源 2 和来源 5 中提到的“莫斯科姆”直接矛盾。

[Thought 2] 我遇到了一个矛盾。来源 3 声称“Davyum”已被批准，而来源 2 和 5 则声称是“Moscovium”。提示中提到 2014 年有一个关于 Davy 的“非官方提案”，这很可能解释了“Davyum”传闻的存在。然而，2016 年 11 月正式批准的名称被广泛记录为 Moscovium。我需要核实“Davy”提案的起源与发现者（很可能是俄罗斯/美国）的命名权之间的关系。

[动作 2] 稠密 _ 搜索 (查询 = “Davyum 提议元素 115 与莫斯科维姆命名权的原点”)

[观测 2] ... 杜布纳团队（俄罗斯）提议以莫斯科地区命名“”……“达维姆”这一名称由网络请愿/活动提出，但并非发现者的选择……2015 年 12 月 30 日，发现获得认可……（搜索结果证实，“”是发现者正式选定的名称）。

[思考 3] 我现在可以解决这个冲突了。

[批判性思维]

Source 3 中关于“Davyum”的片段似乎基于非官方活动的**虚假信息**。它与多个其他可靠来源所关联的 Moscovium 的具体日期（2016 年 11 月 28 日）相矛盾。

题目中提到的是对达维的提议，但在发现者行使命名权后，官方于特定日期（2016 年 11 月）批准的名称是“莫斯科世姆”。

[最终答案] 镆

图 8: 展示对“主观谬误”具备鲁棒性的轨迹。检索系统返回了一个**幻觉或虚假文档**，声称元素 115 被命名为“Davyum”。智能体将该信息与其他来源进行交叉验证，正确识别出该干扰项为源自非官方提案的错误信息，并利用**绿色块**中突出的逻辑进行了推理，从而展示了系统在面对冲突信息和幻觉源时的鲁棒性。

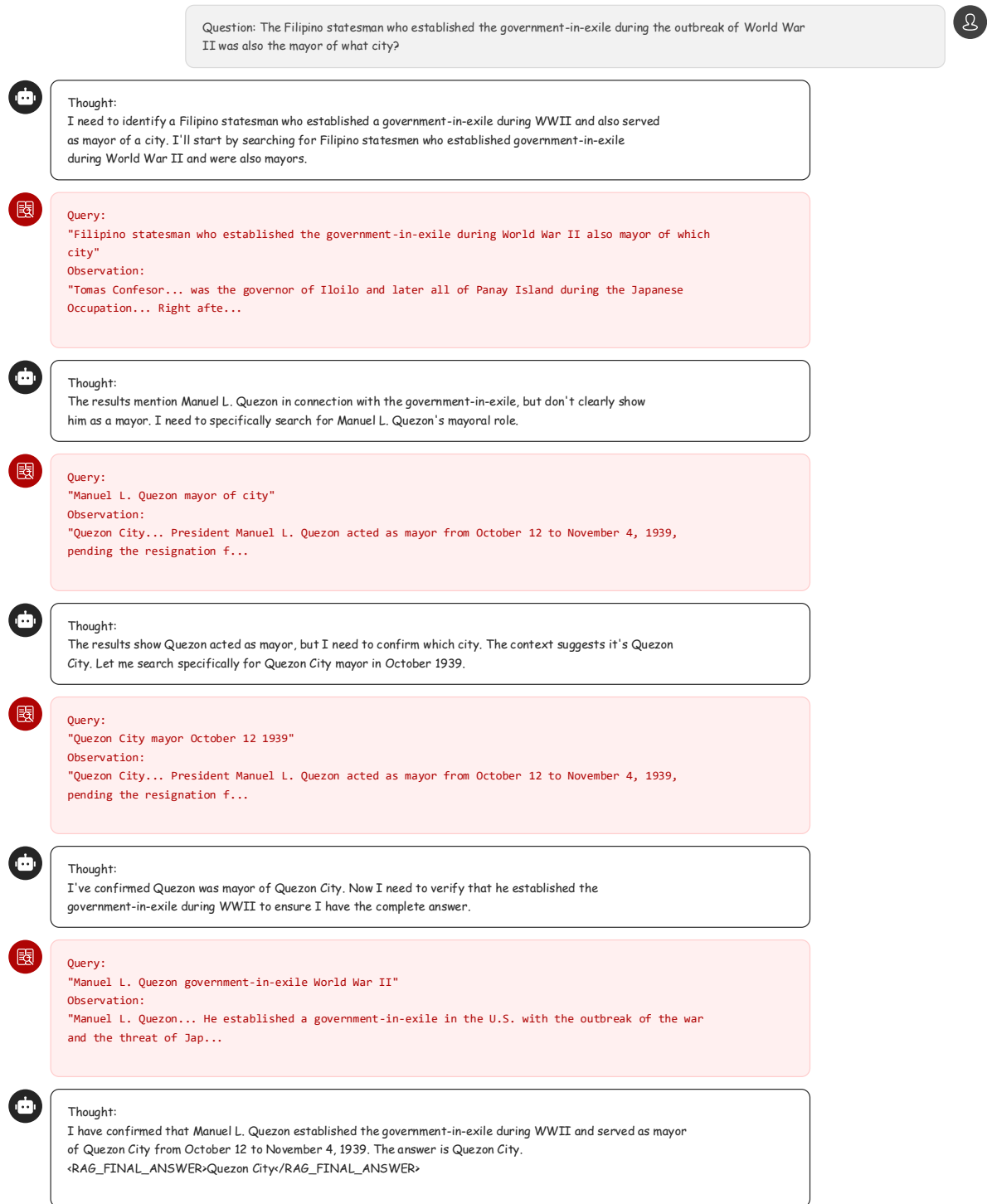


图 9: 由在 Qwen3-30B-A3B-Thinking-2507 上训练的 4.5k RAGShaper 生成的轨迹。该图展示了智能体解决多跳问题的过程：首先，根据“流亡政府”线索识别出可能的候选者（曼努埃尔·L·奎松）；接着，转向验证具体的市长角色（动作 2），发现他曾担任奎松市市长；最后，通过交叉参考二战背景（动作 3）来验证答案。

Tool Schema

稠密搜索

```
{
  "type": "function",
  "function": {
    "name": "query_knowledge_base_dense",
    "description": "[Dense Search] Semantic vector search over the knowledge base. Falls back to configured top_k or 5.",
    "parameters": {
      "type": "object",
      "properties": {
        "query": {
          "type": "string",
          "description": "Natural language question or statement to retrieve against the KB.",
          "minLength": 1
        },
        "top_k": {
          "type": "integer",
          "description": "Override for number of results; positive integer.",
          "minimum": 1
        }
      },
      "required": ["query"]
    }
  }
}
```

图 10: 用于稠密向量检索工具 (*query_knowledge_base_dense*) 的工具模式定义。

Core Prompt of Exploration in Information Curation

=== 主要目標 === 以一條後續將支援 低門檻但深度多跳問答的軌跡。你不僅僅是在收集事實——你是在建立一個依賴鏈 ($A \rightarrow B \rightarrow C \rightarrow D \dots$) 以及容易混淆但可辨別的負樣本文檔。

=== 采样策略 & 规则 ===

1) 构建多跳骨干网络 (深度优先链)

- 目标 \geq 尽可能在 10 个依赖跳数内完成 ($A \rightarrow B \rightarrow C \rightarrow D \dots$)。
- 每次检索步骤必须解锁一个用于下一步的**新**实体/关系。
- 切勿在同一个实体上反复打转。仅在核对关键元数据时返回检查。

2) 每跳打包紧凑证据

- 每跳捕捉 1-2 个简短、可引用的语句，清晰地表达关系。
- 至少捕获一个可交叉核对的**硬元数据项** (年份/日期/版本/编号/数量)。
- 确保最终答案-关键元数据由 ≥ 2 次独立观测支持。

3) 早期且反复生成负样本文档

- **工具使用**: 您必须使用 `write_distractor_docs` (传入 `distractor_texts` 列表)。不要调用大语言模型; 请自行撰写文本。
- **时间**: 在首次成功检索之后以及每次关键跳转时 (尤其是出现困难元数据时) 创建负样本文档。
- **数量**: 最小 ≥ 3 次调用总计; 总计 \geq 每个种子 5 个干扰文档。多样化维度。

4) 安全规则: 必须进行消歧义

- 求解器无法知道哪个文档是干扰项。每个负例文档必须在逻辑上可区分 (例如, 具体年份、版本或范围)。
- 不当: “成立于 2015 年”与“成立于 2016 年”之间缺乏其他上下文。
- 良好: “2015 年年度报告 (审计后)”与“2016 年初稿”。

=== 维度指导 (负面文档类型) ===

- **[A] 复制品**: 相邻版本文档 (例如, 2015 版与 2016 版手册)。仅更改一个规格/数值, 其余保持相似。明确标注版本。
- **[B] 错误捷径**: 一份文档声称 $A \rightarrow C$ (跳过 B) 且使用了模棱两可的措辞, 与真实的 $A \rightarrow B \rightarrow C$ 链路相矛盾。
- **[C] 零碎拼图**: 仅包含部分信息的文档, 从局部看似合理但不完整。
- **[D] 主观谬误**: 带有评论/观点语气, 且包含一个看似合理的事实性错误 (例如, 错误的型号编号)。

图 11: **核心探索轨迹**提示。与标准检索不同, 此提示驱动智能体主动构建深度依赖链 (10+ 跳), 并在滚动过程中合成“复制品”或“虚假捷径”类负样本文档, 为高复杂度谜题生成奠定基础。

Prompt: Trajectory-to-QA Synthesis

请根据轨迹生成一个高质量的问答对：

Question Requirements (Crucial for Reasoning & Brevity): 目标答案必须是一个具体事实（例如，一个名字、一个日期、一个地点、一个数量，或一个是/否状态）。

- ** 不要 ** 提出需要长篇文字解释的“如何”、“为什么”或“描述”类问题。
- ** 反捷径 **：问题不得包含答案文本，且不得以定义性从句直接陈述所询问的属性。
- 低门槛，深推理：问题保持在两句话以内，并提供少量顶层线索；深度应来自多跳依赖链，而非冗长的琐碎信息列表。
- ** 深度多跳（必需）**：问题必须需要至少 3 个相互依赖的步骤才能解决（仅链式依赖；无星型检查清单）。
- ** 负文档混淆性（必需）**：如果轨迹包含负文档（例如通过 `write_distractor_docs` 生成），请设计问题，使得粗心的求解者可能被至少一个负文档误导，从而得出一个看似合理的错误答案/路径，而正确答案仍能得到轨迹中权威证据的支持。- 问题应为自然、事实性且自包含的问题（例如，不要包含“智能体发现了什么...”、“轨迹中有什么...”、“根据轨迹...”等表述）。它必须看起来完全不像前一步进行了轨迹探索。且不要提及“搜索”或“搜索结果”之类的内容。

Answer Requirements (Crucial for Strict Length): - ** 极度简短 **：答案必须为 ** 不超过一句话，且仅包含一个实体 **，或理想情况下仅为一个 ** 短语 **（例如，“1985”，“凡尔赛条约”，“增加 5%”）。

- ** 无冗余信息 **：不要使用“根据文件...”或“答案是...”之类的填充词。仅提供最终的答案值。
- 事实性：特定事实必须严格源自提供的轨迹观测，不得提及轨迹或观测。

Required Explanations (for dataset traceability; NOT part of the question text):

- `reasoning_steps`: 提供 ≥ 3 简短且相互依赖的步骤，仅使用轨迹证据来解决问答问题。
- `negative_aspect`: 解释负向文档可能如何产生误导，以及何种消歧措施能够克服这些误导。在可能的情况下提及干扰维度。

消歧义：如何澄清误导性说法。

- `distraction_text`: 用于分散求解者注意力的文本。

确切地按照此模式返回 JSON（不要添加额外字段）：

```
{
  "question": "question text",
  "answer": "short phrase or single sentence",
  "reasoning_steps": [
    {"hop": 1, "fact": "intermediate fact", "evidence": "snippet", "output": "entity/metadata"},
    {"hop": 2, "fact": "intermediate fact", "evidence": "snippet", "output": "entity/metadata"},
    ...
    {"hop": n, "fact": "final derivation", "evidence": "snippet", "output": "answer"}
  ],
  "negative_aspect": [
    {"dimension": "doppelganger|false_shortcut|fragmented_puzzle|subjective_fallacy",
      "misleading_claim": "claim", "disambiguation": "method", "distraction_text": "text"}
  ]
}
```

图 12: **QA 合成**提示。此提示消耗上一步生成的轨迹。它施加严格的约束，以确保合成的问题具有“低门槛”（简洁）特性，同时具备“深度推理”（需要遍历完整的依赖链）特性，并显式验证负样本文档的有效性。

Prompt for Trajectory Rollout

请提供具体问题，以便我使用稠密检索系统进行查询和验证。

Core Capabilities

- 语义理解：系统匹配您的查询的含义，而不仅仅是确切的词语。
- ** 处理改写内容 **：即使使用不同的术语，也能找到相关内容。

Query Formulation Strategy

1. ** 描述要具体 **：编写完整描述你需求的自然语言查询。- * 不好 *：“2023 年收入” - * 好 *：“该公司在 2023 财年的总营收是多少？”
2. ** 上下文很重要 **：在查询字符串中包含必要的上下文信息，因为检索器会独立处理每个查询。
3. ** 迭代优化 **：- 如果结果过于宽泛：在查询中添加具体的约束条件。- 如果结果不相关：使用同义词或相关概念重新表述查询。

Execution Protocol

1. 将复杂的多跳问题分解为分离的、更简单的查询。
2. 确认检索到的内容是否符合用户意图。
3. 如果经过多次尝试（超过 5 次）仍找不到相关信息，请尝试用不同的方法重新表述您的问题。

Internal Knowledge Fallback Mechanism

当您在多轮尝试检索查询后仍无法在知识库中找到答案时，应使用您的内部知识提供尽可能最佳的答案。这是确保即使知识库中没有所需信息，您仍能帮助用户的一种备用机制。使用内部知识时，请在推理过程中明确说明，并将您的答案用最终答案标签包裹。

Critical Requirements

1. ** 推理工具一致性 **：如果您的推理过程中提到了使用工具（例如“让我们搜索”、“我们需要使用稠密检索工具”），您必须生成相应的 `tool_calls`。不得仅停留在推理阶段。
2. ** 动作跟进 **：如果你决定在推理过程中使用某个工具，就必须执行实际的工具调用。仅包含关于工具使用的想法而没有具体内容的回复不是有效的最终答案。

Answer Strategy

答案：简短回答

{FINAL_ANSWER_START}2. ** 必填 **：您必须将最终答案包裹在 {FINAL_ANSWER_START} 和 {FINAL_ANSWER_END} 标签中。未使用这些标签提供答案是不允许的。每个包含答案的回复都必须使用这些标签。{FINAL_ANSWER_END}

具体实体：姓名、地点、数字、日期、编号或其他具体信息。

- ** 禁止 ** 使用“和”、“或”、“的”、“在”、“是”、“曾”、“为”、“有”、“一个”、“一些”、“作为”、“为了”、“与”、“从”、“到”、“上”、“在”、“通过”、“这个”、“那个”、“这些”、“那些”等常见词汇作为最终答案
- 常见词汇、冠词、介词和连词不是有效答案。答案应为能直接回答问题的有意义实体或信息。

{FINAL_ANSWER_START} 如果检索到的信息中没有明确的答案，请说明无法找到答案，但仍需将回答内容用答案标签包裹。{FINAL_ANSWER_END}

图 13: 在 **轨迹滚动**阶段用于引导智能体生成训练数据的完整提示。它明确指示模型在查询构造策略、备用机制以及最终答案所需严格格式方面的要求。

Prompt for Evaluation

请使用稠密检索系统（语义/向量搜索）来回答和验证问题。必须使用稠密检索工具，不要尝试使用稀疏检索工具，因为它们不可用。

Core Capabilities

- 语义理解：系统匹配您的查询的含义，而不仅仅是确切的词语。
- ** 处理改写内容 **：即使使用不同的术语，也能找到相关内容。

Query Formulation Strategy

1. ** 描述要具体 **：编写完整描述你需求的自然语言查询。- * 不好 *：“2023 年收入” - * 好 *：“该公司在 2023 财年的总营收是多少？”
2. ** 上下文很重要 **：在查询字符串中包含必要的上下文信息，因为检索器会独立处理每个查询。
3. ** 迭代优化 **：- 如果结果过于宽泛：在查询中添加具体的约束条件。- 如果结果不相关：使用同义词或相关概念重新表述查询。

Execution Protocol

1. 将复杂的多跳问题分解为分离的、更简单的查询。
2. 确认检索到的内容是否符合用户意图。
3. 如果经过多次尝试（> 5）后仍找不到相关信息，则承认该信息在知识库中缺失。

Answer Strategy

答案：简短回答

<RAG_FINAL_ANSWER>2. 将最终答案包裹在 <RAG_FINAL_ANSWER> 和 </RAG_FINAL_ANSWER> 之间，并将任何推理保留在标记之外。</RAG_FINAL_ANSWER>

Available Tools

- query_knowledge_base_dense: [稠密搜索] 在知识库上进行语义向量搜索。若无结果，则回退到配置的 top_k 或 5。

图 14: 评估阶段使用的提示。与训练提示相比，此版本指导模型优先考虑诚实，当知识库中缺少信息时承认这一点，而不是依赖内部知识。它还指定了用于最终答案提取的 XML 样式标签。