Eliciting Sophisticated Agentic RAG Skills via Automated Data Synthesis

Anonymous ACL submission

Abstract

Agentic Retrieval-Augmented Generation (RAG) empowers large language models to autonomously plan and retrieve information for complex problem-solving. However, the development of robust agents is hindered by the scarcity of high-quality training data that reflects the noise and complexity of real-world retrieval environments. Conventional manual annotation is unscalable and often fails to capture the dynamic reasoning strategies required to handle retrieval failures. To bridge this gap, we introduce RAGShaper, a novel data synthesis framework designed to automate the construction of RAG tasks and robust agent trajectories. RAGShaper incorporates an `Information Curator` to build dense information trees enriched with adversarial distractors spanning *Perception* and *Cognition* levels. Furthermore, we propose a constrained navigation strategy that forces a teacher agent to confront these distractors, thereby eliciting trajectories that explicitly demonstrate error correction and noise rejection. Comprehensive experiments confirm that models trained on our synthesized corpus significantly outperform existing baselines, exhibiting superior robustness in noise-intensive and complex retrieval tasks.

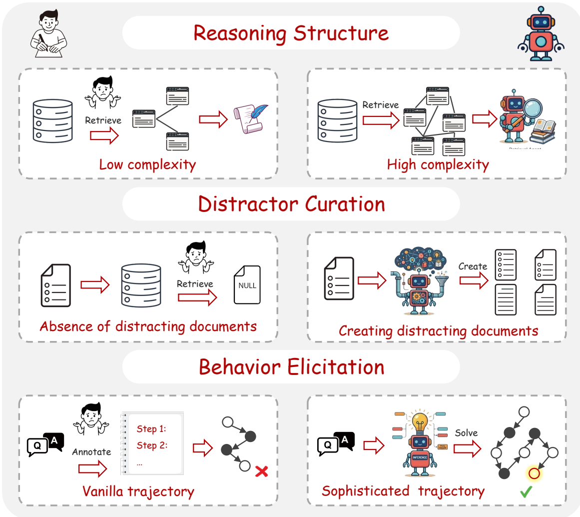

Figure 1: Limitations of human annotation of the agentic RAG dataset, which can be mitigated by automatic synthesis by the agent curator.

1 Introduction

Agentic Retrieval-Augmented Generation (Agentic RAG) has emerged as a pivotal advancement in natural language processing, rapidly evolving from simple retrieval-and-read pipelines to autonomous systems capable of complex reasoning and dynamic tool usage (Jin et al., 2025; Asai et al., 2024; Li et al., 2025a; Team et al., 2025). As Large Language Models (LLMs) are increasingly deployed in open-ended environments, Agentic RAG serves as the core infrastructure for a wide array of sophisticated applications, ranging from autonomous research assistants to domain-specific decision support systems. By endowing models with the agency to actively plan retrieval steps, evaluate gathered information, and iteratively refine their search, this paradigm represents a significant leap forward in bridging the gap between static knowledge bases and intelligent responses (Singh et al., 2025).

Current methodologies predominantly rely on manually annotated datasets, typically structured as question-trajectory-answer triplets (Yang et al., 2018; Ho et al., 2020). However, this paradigm is fundamentally ill-suited for training Agentic RAG models due to the intrinsic cognitive and operational bottlenecks of human annotators, as shown in Figure 1. **First**, constrained by limited working memory, annotators struggle to synthesize implicit, multi-hop evidence scattered across a large volume of disparate documents, often defaulting to shallow, single-context reasoning rather than the deep retrieval chains required for robust agents (Wu et al., 2025). **Second**, manually curating realistic, noise-heavy retrieval environments is impractical. Retrieval distractors that are lexically similar yet factually incorrect may not exist (Yan et al.). **Fi-**

nally, human annotations is hard to capture the dynamic strategy adjustments required to decompose tasks and recover from retrieval failures (Jeong et al., 2024; Tian et al., 2025). Consequently, these limitations make the manual construction of high-quality data for Agentic RAG difficult to scale.

To surmount these impediments and automate the production of high-fidelity training corpora, we introduce **RAGShaper**, a novel framework specifically engineered for Agentic RAG data synthesis. Addressing the complexity of information construction, RAGShaper incorporates an `InfoCurator` module designed to autonomously build a comprehensive retrieval environment. Starting from a seed entity, the curator leverages retrieval tools to perform multi-round exploration within the knowledge base, aggregating a dense information tree of entities and interrelations to support the synthesis of tasks requiring deep reasoning. Beyond gathering positive evidence, the curator dynamically generates adversarial "distractor" documents based on the retrieved context. We systematically categorize these distractors into two dimensions, *Perception* and *Cognition*, a taxonomy designed to cultivate robust agentic discrimination capabilities against varying levels of information noise. Following information curation, an LLM utilizes this structured context to synthesize specific tasks and corresponding ground-truth answers. To extract optimal skill and behavior patterns, we employ a sophisticated teacher agent to solve these synthesized tasks; uniquely, we enforce a constrained navigation strategy that mandates the retrieval of the generated distractors, thereby explicitly capturing the teacher's adaptive strategies in identifying and overcoming information hazards. Finally, by fine-tuning a base model on this large-scale corpus of agent trajectories, we obtain a robust Agentic RAG model proficient in navigating noisy environments.

We summarize our contributions as follows:

- We introduce RAGShaper, an agentic RAG data synthesis framework featuring an `InfoCurator` designed to aggregate densely connected information and synthesize sophisticated retrieval distractors across multiple dimensions.
- We propose a constrained navigation strategy to elicit robust error-correction and reasoning behaviors from the teacher agent, enabling the large-scale accumulation of high-quality, resilient trajectories.
- We conduct extensive experiments to validate our data synthesis framework, with empirical results demonstrating that models trained on our corpus significantly outperform baselines in complex retrieval environments.

2 Preliminaries

We formalize the Agentic RAG framework as an autonomous agent that interleaves reasoning with retrieval, enabling to dynamically interact with external corpora to resolve knowledge-intensive queries. Adopting the ReAct paradigm (Yao et al., 2023), the agent navigates a sequential decision process where it must iteratively bridge the gap between its internal knowledge and the required external evidence. At each time step t, the agent conditions on the initial query and the history of prior interactions to generate a reasoning thought τ_t. This reasoning guides the selection of a specific retrieval tool-use action α_t, such as querying a knowledge base $\mathbb{K}$ to retrieve documents $\mathbb{D}$, which yields a corresponding observation o_t. This cumulative reasoning-retrieval loop is represented by the *agent trajectory*, denoted as:

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

where $\mathcal{Q}$ represents the user task, and the tuple (τ_i, α_i, o_i) captures the agent's planning, tool-use action, and feedback at step i. $\mathcal{A}$ denotes the final answer for $\mathcal{Q}$, representing the agent's primary objective. The purpose of our data synthesis is to construct $(\mathcal{Q}, \mathcal{A}, \mathcal{T})$ triples for RAG agent training.

3 Method

We propose RAGShaper, a data synthesis framework designed to automate the construction of high-quality training corpora for Agentic RAG. As illustrated in Figure 2, our pipeline consists of four phases: (1) **Information Curation** (§3.1), where an autonomous curator agent explores a seed entity to build a dense, distractor-augmented information tree, followed by a selection process to identify useful information paths; (2) **Question-Answer Synthesis** (§3.2), where tasks are derived from these selected paths; (3) **Behavior Elicitation** (§3.3), where a teacher agent solves these tasks under a specific distraction strategy to generate trajectories exhibiting sophisticated behaviors; and (4) **Training** (§3.4), where the student model is fine-tuned on these enhanced trajectories.

3.1 Information Curation

To train agents capable of deep reasoning, the underlying information retrieval tasks must be rich

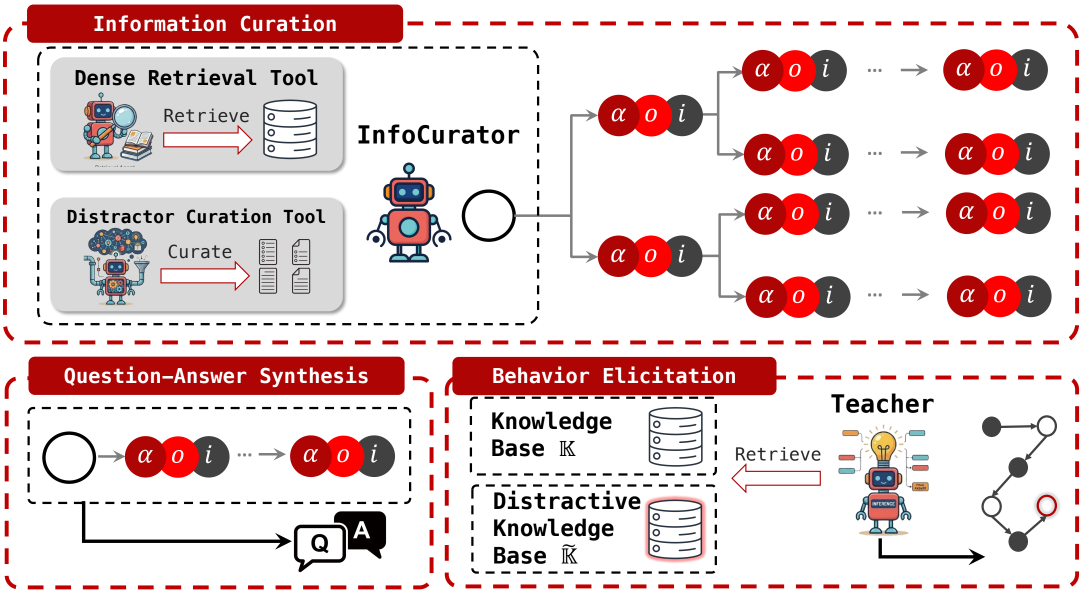

Figure 2: Overview of RAGShaper.

in inter-entity relationships and semantically challenging noise. As manually constructing such information structures is not scalable, we design an `InfoCurator` agent to automate this process.

3.1.1 Tree Exploration on `InfoCurator`

The goal of the `InfoCurator` is to construct an information structure from a knowledge base $\mathbb{K}$, which serves as the foundation for subsequent question synthesis. Specifically, `InfoCurator` builds an information tree by retrieving positive facts and crafting distractive documents. The exploration begins with a seed entity, which serves as the root node s_1 of the tree. `InfoCurator` then expands new nodes via depth-first traversal to explore new information. A node is defined as:

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

where α_t and i_t represent the action and intention of `InfoCurator` for expanding the node. The action α_t involves either retrieving documents or creating distractive ones (detailed below), and o_t is the observation resulting from α_t. We crawl large-scale entities from Wikipedia[1]. The agent expands a node by invoking tools based on the path from the current node to the root:

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

At each step, we expand two child nodes with probability p^b and one node with probability $1 - p^b$. Expansion terminates when the node depth reaches a predefined threshold. The resulting information tree contains facts and their relations. This automated process significantly alleviates the workload of manual data organization. We detail the tools used by `InfoCurator` below.

Dense Retrieval Tool. `InfoCurator` is equipped with a Dense Retrieval Tool. The parameters include a *Query* and *Topk*, representing the search string and the desired number of relevant documents, respectively. The tool encodes the query using a pretrained text embedding[2] and computes the similarity between the query and documents indexed in the KB. It returns documents with similarity scores exceeding a threshold τ, ensuring the output count does not exceed k:

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

where d represents a document and τ denotes the similarity threshold.

Distractor Curation Tool. A robust Agentic RAG model must distinguish between relevant evidence and noise. Merely including positive facts in the information set is insufficient; we must also include challenging distractors. However, relying solely on retrieving similar facts from the KB as distractors is often impractical due to lack of precision or the absence of suitable candidates. Therefore, we introduce the Distractor Curation Tool, which directly generates and stores distractive documents.

[1] https://www.wikipedia.org/

[2] We use E5 as the retriever in the DPR project: 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.
Cognition Layer	False Shortcut	Forged A→C direct connection (real logic: A→B→C) with ambiguous/wrong justifications.	Truth: Virus → **Fever** → Weakness. Distractor: "Whether the virus causes weakness remains **unknown**".	***Reasoning Persistence***: Reject shortcuts; search for intermediate nodes.
	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.

Table 1: Distractor types, examples, and corresponding target agent skills.

A distractive document is not necessarily factually incorrect but is designed to be confusing within the context of the RAG task. We include four types of distractors spanning both *Perception* and *Cognition* levels, as shown in Table 1. The tool takes an *Original fact*, a *Distractor type*, and a *Creating guideline* as input, calling an LLM to generate a distracting fact based on these parameters. The guideline ensures the generated content is precise.

3.1.2 Information Path Selection

After building the information structure, we identify specific sub-structures for QA synthesis. The raw tree contains numerous divergent paths, not all of which form coherent reasoning chains. We employ a heuristic selection mechanism to extract high-value paths from the root to the leaves. We posit that a desirable path contains a high density of information. Thus, we score each path based on the total number of documents it contains, including both positive entries and distractors:

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

where $|\mathcal{D}_s|$ denotes the document count at node s, and s_l is a leaf node. We select the m paths with the highest scores for synthesis.

3.2 Question-Answer Synthesis

Once the paths are selected, we synthesize the task $(\mathcal{Q}, \mathcal{A})$. Motivated by the need to align the question with the retrieval steps, we prompt an LLM to "reverse-engineer" a question that strictly requires the information sequence found in the path to answer. The generator conditions on the full sequence of observations and intents:

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

Here, the inclusion of the intent i is critical. By explicitly exposing the `InfoCurator`'s intent, the LLM ensures that $\mathcal{Q}$ naturally necessitates that specific information, guaranteeing that the path serves as a valid reasoning support.

3.3 Behavior Elicitation

After harvesting the $(\mathcal{Q}, \mathcal{A})$ pairs, we construct the agent execution trajectory $\mathcal{T}$. Directly using the curated path is suboptimal, as it may be noisy or not represent the most efficient solution. Instead, we employ a `Teacher` agent to solve $\mathcal{Q}$, thereby generating the final training trajectory $\mathcal{T}$:

$$\tilde{\mathcal{A}}, \mathcal{T} = \texttt{Teacher}(\mathcal{Q}), \tag{7}$$

where $\tilde{\mathcal{A}}$ is the predicted answer. The trajectory follows the format defined in Eq. (1). The teacher agent is equipped only with the Dense Retrieval Tool, identical to `InfoCurator`.

To elicit the sophisticated behaviors and abilities outlined in Table 1, we implement a specific strategy using the generated distractors. We aggregate all distractive documents into a secondary KB, $\tilde{\mathbb{K}}$. During retrieval, the tool fetches documents $\mathbb{D}_t$ from both the original KB and $\tilde{\mathbb{K}}$ according to the following logic:

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

where p^e is a fixed probability. Specifically, at the first step, the agent is forced to retrieve from $\tilde{\mathbb{K}}$. If retrieval from $\tilde{\mathbb{K}}$ occurred in the previous step, it is suppressed in the current step to prevent continuous hallucination loops. Otherwise, retrieval from $\tilde{\mathbb{K}}$ occurs with probability p^e. Crucially, the `Teacher` agent remains agnostic to the existence of $\tilde{\mathbb{K}}$.

3.4 Training

Finally, we compile the synthesized triples $(\mathcal{Q}, \mathcal{A}, \mathcal{T})$ into a training dataset, retaining only trajectories where the predicted answer is correct (i.e., $\tilde{\mathcal{A}} = \mathcal{A}$). We fine-tune a base LLM to minimize the standard negative log-likelihood loss on the agent trajectory tokens, following standard supervised fine-tuning (SFT) protocols:

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

where x_i is the ith token and $\mathbb{I}$ is the indicator function masking observation tokens. By training on trajectories $\mathcal{T}$ that include behaviors such as self-correction and distractor rejection, derived from our constrained elicitation process, the resulting model learns to operate autonomously in noisy, open-ended retrieval environments.

4 Experiments

4.1 Experimental Settings

Data Synthesis. We set the branch probability p^b to 0.5 if it's on the first 2 depth of the exploration tree, otherwise $p^b = 0$. The maximum depth of the tree is 30. The dense retrieval tool threshold τ is 0.8. The distractive probability p^e in Behaviour Elicitation is 0.5. We select two paths ($m = 2$) for data synthesis from each exploration tree. We use gpt-oss-120b as the Teacher agent, where `InfoCurator` is based on it as well.

Training. We train on Qwen3-30B-A3B-Think and Qwen3-4B-Think (Team, 2025) on Megatron-LM framework. We use 4.5k and 6k data settings. Details are in the Appendix A.

Evaluation Benchmark. To comprehensively evaluate the reasoning and retrieval capabilities of our agent, we conduct experiments on four diverse open-domain RAG benchmarks: Natural Questions (NQ) (Kwiatkowski et al., 2019), PopQA (Mallen et al., 2023), AmbigQA (Min et al., 2020), and Bamboogle (Press et al., 2023). We report performance using standard Exact Match (EM) and F1 Score metrics. We use evaluation setting the same as DecEx-RAG. Details are in Appendix B.

Baselines. We compare RAGShaper against a wide range of competitive baselines. For prompt-based methods, we include 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), and Search-o1 (Li et al., 2025a). Regarding learning-based methods, we benchmark against DeepRAG (Guan et al., 2025), IKEA (Huang et al., 2025), ReasonRAG (Zhang et al., 2025), DecEx-RAG (Leng et al., 2025), Search-R1 (Jin et al., 2025), and HL-Data (Jin et al., 2025; Leng et al., 2025) (i.e. Subset of HotPotQA and 2Wiki, therefore we don't take them for evaluation). Detailed descriptions are provided in Appendix C.

4.2 Main Results

RAGShaper establishes significant improvement. Table 2 presents the comparison of RAGShaper against state-of-the-art baselines. Our method consistently achieves the best performance, with the 6.5k model setting a new benchmark of 50.3 Avg EM and 62.0 Avg F1, significantly surpassing both prompt-based (e.g., Search-o1) and learning-based methods.

Synthesized data surpasses human annotation in quality. Crucially, RAGShaper demonstrates superior data efficiency compared to human annotation. Under the same data scale (4.5k), our method outperforms HL-Data across almost all metrics. This indicates that our automated pipeline generates higher-quality training data which excels traditional crowd-sourced data.

Distractor training enables robustness on complex, noisy tasks. The performance gains are most pronounced on complex, noise-intensive tasks like Bamboogle and AmbigQA. The significant lead on AmbigQA directly validates the effectiveness of our *Distractor Curation* mechanism and *Behaviour Elicitation*. By training on trajectories laden with multi-dimension of distractors, our agent effectively learns to filter retrieval noise and execute robust multi-hop reasoning, a capability essential for navigating the ambiguity inherent and adapting retrieving strategy in these challenging datasets.

4.3 Ablation Study

To assess the contribution of our distractor-based learning mechanism, we conduct an ablation study using a variant named RAGShaper-Dis. We exclude the Distractor Curation Tool during data synthesis and remove the noise-injection strategy during the Behavior Elicitation phase. The agent is trained solely on clean, positive reasoning paths without exposure to adversarial retrieval contexts.

Models	Bamboogle		PopQA		NQ		AmbigQA		Avg	
	EM	F1	EM	F1	EM	F1	EM	F1	EM	F1
Prompt-Based Methods										
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
Learning-Based Methods										
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
Ours										
RAGShaper 4.5k	58.5	70.3	37.4	47.8	38.3	50.0	**61.3**	**71.4**	48.8	59.8
RAGShaper 6.5k	**60.0**	**72.6**	38.9	49.6	**41.3**	**54.8**	61.1	71.1	**50.3**	**62.0**

Table 2: Performance comparison on evaluation datasets. HL-Data denotes open-sourced human labeled data, i.e. sampled HotpotQA and 2WikiMultiHopQA from training set. Avg is recalculated based on Bamboogle, PopQA, NQ, and AmbigQA. **Bold** stands for the highest score, and underline is the second best.

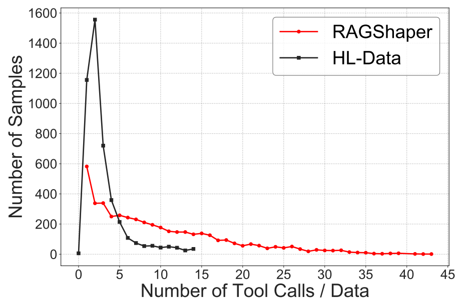

Figure 3: Tool call statistics on 4.5k data on both RAGShaper and HL-Data.

Distractor-based learning is essential for robust retrieval. As shown in Table 3, removing these components leads to a severe performance drop, with the Average EM plummeting from 48.8 to 33.8. The decline is most dramatic on noise-sensitive datasets like AmbigQA and Bamboogle. These results strongly underscore the necessity of our approach: training on "clean" data alone is insufficient for robust agentic retrieval. The proposed synthesis of perception and cognition-level distractors is essential for equipping the agent with the critical ability to discern evidence from noise in complex real-world environments.

4.4 Trajectory Complexity Analysis

To further investigate the reasoning quality of our synthesized corpora, we analyze the distribution of tool usage steps per trajectory. In Figure 3, we compare the trajectory depth of RAGShaper against the human-labeled baseline (HL-Data).

RAGShaper synthesizes deeper, more complex reasoning tasks. The distribution reveals a significant distinction in task complexity. HL-Data exhibits a sharp peak at 2-3 steps with a short tail, indicating that most human-annotated samples represent relatively shallow, few-shot reasoning tasks. In contrast, RAGShaper presents a much broader, long-tailed distribution, with a substantial portion of trajectories requiring over 10, and up to 40+, steps. This confirms that our method successfully synthesizes tasks of higher difficulty.

Longer trajectories encode richer agentic behaviors. Crucially, a higher number of tool calls implies a richer density of agentic behaviors. The long-tail trajectories in RAGShaper capture complex cognitive processes, such as navigating dead ends, verifying distractors, and performing extensive multi-hop planning, that are rarely present in the concise HL-Data. Furthermore, unlike generic datasets where models might answer from parametric memory, our distribution starts strictly after zero, ensuring that every trajectory involves necessary retrieval actions. This eliminates trivial "direct answer" cases and enforces a rigorous evidence-seeking process.

Models	Bamboogle		PopQA		NQ		AmbigQA		Avg	
	EM	F1	EM	F1	EM	F1	EM	F1	EM	F1
Qwen3-30B-A3B-Think										
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
Qwen3-4B-Think										
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

Table 3: Ablation studies and experiments on different backbones. RAGShaper-Dis stands for experiments on distractive documents created and added in the Behaviour Elicitation.

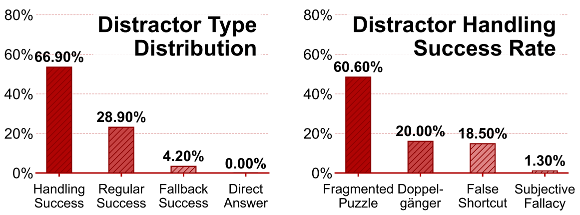

Figure 4: Trajectory analysis.

4.5 Trajectory Behavior Analysis

To understand the underlying mechanisms of our model's success, we analyze the distribution of agent behaviors within the synthesized trajectories. We use LLM to tag each trajectory types, results are as visualized in Figure 4 (Left).

Agents rely on rigorous retrieval rather than internal knowledge. The analysis reveals that the majority of trajectories (66.90%) are categorized as *Handling Success*, where the agent successfully identifies and resolves the injected distractors to reach the correct answer. This high proportion, when viewed in conjunction with the high number of tool calls observed in Section 4.4, confirms that our dataset is rich in high-quality agentic behaviors. The agent is not merely retrieving; it is actively reasoning against noise. Furthermore, the results indicate a strict reliance on retrieval rather than internal knowledge. The *Direct Answer* rate is 0.00%, and *Fallback Success* (answering correctly despite failing to retrieve useful information) comprises only 4.20%. This low prevalence of non-retrieval based answers demonstrates that the performance improvements are driven by the agent's enhanced ability to interact with external corpora, rather than by internal knowledge hallucinations or simple memorization.

Complex cognitive traps provide headroom for future improvement. Figure 4 (Right) further dissects the success rates across different distractor types tagged by LLM, revealing a distinct hierarchy of difficulty. While the agent shows competence in solving *Fragmented Puzzles* (60.60%), which primarily tests information aggregation, it encounters significant challenges with deeper cognitive traps. The low success rates for *False Shortcut* (18.50%) and the extremely challenging *Subjective Fallacy* (1.30%) suggest that the upper bound of our data's difficulty has **not** yet been reached. This "headroom" indicates that RAGShaper provides a sufficiently complex environment for further research. Future work could leverage this unexploited complexity through advanced training paradigms, such as reinforcement learning, to enable agents to master these subtle and adversarial reasoning scenarios.

4.6 Generalization on Different Backbones

To further verify that the effectiveness of RAGShaper is not limited to a specific model architecture, we extended our evaluation to a different backbone: Qwen3-4B-Think. We compare the performance of models fine-tuned on our synthesized data against those trained on HL-Data of the same scale (4.5k). The results are summarized in Table 3.

RAGShaper demonstrates strong generalization across diverse backbones. As observed in the table, RAGShaper consistently outperforms the HL-Data baseline, achieving a significant improvement in the overall average score. This confirms that the high-quality reasoning trajectories generated by our pipeline are universally beneficial and transferrable, rather than being overfitted to the specific characteristics of the certain experimental model.

4.7 Case Study

Figure 5 shows a QA case with its distracting documents. We add reasons why these distracting documents can elicit sophisticated behaviours. Our method can generate various and effective distractors to stimulate advanced abilities of RAG agent.

5 Related Work

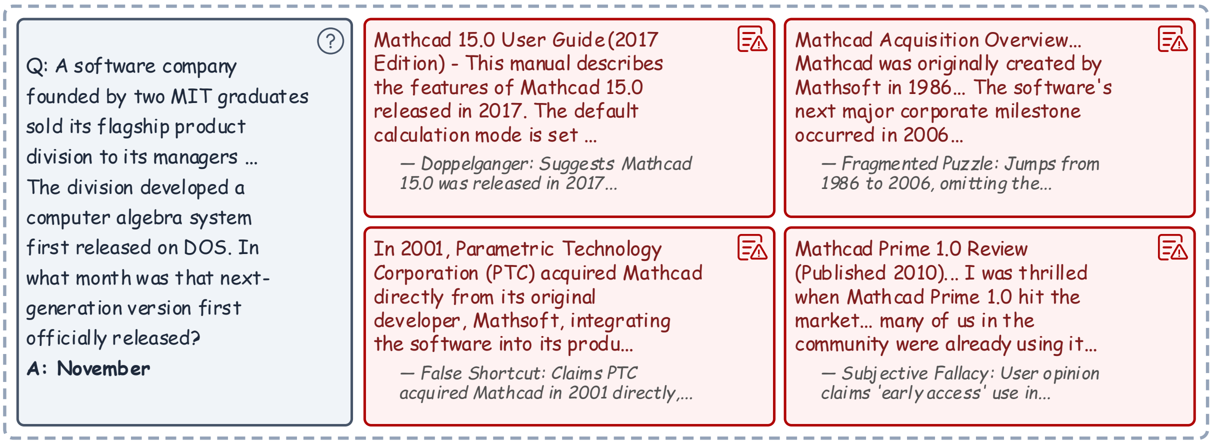

Figure 5: An illustrative example of the distractor taxonomy used in data synthesis. The figure visualizes four distinct categories of cognitive traps (Doppelganger, Fragmented Puzzle, False Shortcut, and Subjective Fallacy) designed to challenge the agent's retrieval and reasoning robustness.

Retrieval-Augmented Reasoning Methods. Existing work improves RAG through both prompt-based and learning-based approaches. Prompt-based methods enhance inference without updating model parameters, including interleaving retrieval with chain-of-thought reasoning (Shao et al., 2023; Trivedi et al., 2023), triggering retrieval adaptively based on generation confidence (Jiang et al., 2023), and compressing context to improve information efficiency (Li, 2023; Jiang et al., 2024; Xu et al., 2023; Lee et al.). More recently, proprietary systems such as Search-o1 integrate retrieval tools directly into the reasoning process and achieve competitive performance (Li et al., 2025a; Sun et al.). Learning-based approaches further improve performance by training agents to coordinate retrieval and generation, often formulating the process as a Markov Decision Process (Guan et al., 2025; Huang et al., 2025) or applying process-supervised reinforcement learning for fine-grained optimization (Zhang et al., 2025; Leng et al., 2025). In addition, strong open-weight models like Search-R1 equip reasoning backbones with trainable search capabilities (Jin et al., 2025).

Data for RAG. High-quality RAG systems often rely on human-labeled supervision. Standard baselines (Jin et al., 2025; Leng et al., 2025; Yu et al., 2024; Li et al., 2025b) utilize datasets like HotpotQA and 2WikiMultiHopQA (Yang et al., 2018; Ho et al., 2020), which are manually curated to test multi-hop reasoning. However, constructing such datasets requires labor-intensive annotation to verify evidence chains, making them expensive and difficult to scale for general-purpose training.

Agentically Data Synthesis. To address the data scarcity in training generalist agents, recent studies have pivoted towards agentic data synthesis, where agents are employed to generate high-quality training samples (Gao et al., 2025; Chen et al., 2025; Zhai et al., 2025). Auto-Explorer (Guo et al., 2025) introduces an explorer agent that autonomously navigates and parses GUI environments to collect diverse state-action pairs without human intervention. Similarly, OS-Genesis (Sun et al., 2025) proposes a reverse task synthesis pipeline, where agents first interact with the environment to create trajectories, which are then retrospectively aligned with synthesized high-level instructions. For search agents, WebShaper (Tao et al., 2025) utilizes a formalization-driven framework with an agentic Expander to iteratively generate complex queries and reasoning paths. Furthermore, DeepSeek-V3.2 (Liu et al., 2025) implements a large-scale synthesis pipeline, deploying specialized agents to construct and verify tasks across various domains, enhancing agent generalization.

6 Conclusion

We presented RAGShaper, a framework designed to overcome the scalability and quality limitations of human annotation for Agentic RAG. By leveraging the `InfoCurator`, we automate the construction of dense retrieval environments populated with adversarial distractors across *Perception* and *Cognition* dimensions. Furthermore, our constrained navigation strategy effectively captures robust error-correction behaviors from teacher agents. Empirical results demonstrate that models trained on our synthesized corpus significantly outperform baselines in complex settings.

Limitations

In this work, we leverage RAGShaper to construct sophisticated behaviours of RAG agent. However, as discussed in Section 4.5, our data has not fully unlocked its potential. In future work, more adavanced approaches can be applied to our data with further training mechanisms.

Ethical Considerations

This work uses publicly available wikipedia documents and entities. It won't contain any information that names or uniquely identifies individual people or offensive content. We only use AI for writing assistant.

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-r1: 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 compositionality 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. Os-genesis: 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: Agentically 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, Niranjan 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 Training Details

We fine-tune the Qwen3-30B-A3B-Think[3] and Qwen3-4B-Think[4] models using the Megatron-LM framework. We extend the context length to 128k. We employ the AdamW optimizer with a precision-aware configuration, coupled with a cosine decay learning rate scheduler. This scheduler features a peak learning rate of 1.0×10^{-5}, a minimum learning rate of 1.0×10^{-6}, and a 5% warmup phase. The global batch sizes are configured as 16 for Qwen3-30B-A3B-Think and 40 for Qwen3-4B-Think. Both models are trained for 5 epochs, and the checkpoint exhibiting the best performance is selected for evaluation.

A.1 Evaluation Metrics

Following standard open-domain Question Answering protocols, we employ two primary metrics:

- **Exact Match (EM):** Measures the percentage of predictions that match one of the ground-truth answers exactly after normalization.
- **F1 Score:** Measures the token overlap between the predicted answer and the ground truth, providing a granular assessment of partial correctness.

B Evaluation Benchmarks

We utilize four datasets to evaluate distinct aspects of retrieval and reasoning:

- **Natural Questions (NQ) (Kwiatkowski et al., 2019):** A large-scale benchmark comprising real user queries issued to Google Search. We utilize the open-domain split, requiring the agent to retrieve answers from the entire Wikipedia corpus.
- **PopQA (Mallen et al., 2023):** Designed to evaluate factual retrieval for long-tail entities. This dataset contains queries where parametric memory typically fails, thereby necessitating precise external retrieval.
- **AmbigQA (Min et al., 2020):** Derived from NQ, this dataset focuses on ambiguous queries with multiple plausible answers. It challenges the agent's ability to disambiguate user intent and navigate noisy retrieval contexts.
- **Bamboogle (Press et al., 2023):** A "google-proof" dataset crafted to test multi-hop reasoning. Questions require synthesizing information from multiple distinct documents rather than locating a single direct answer.

C Baseline Details

We compare our approach against the following competitive baselines:

Prompt-Based Methods. These utilize fixed LLMs with advanced prompting or retrieval strategies:

- **Iter-RetGen (Shao et al., 2023):** Iteratively synergizes retrieval and generation, utilizing model outputs to refine subsequent retrieval queries.
- **IR-CoT (Trivedi et al., 2023):** Interleaves chain-of-thought reasoning with retrieval steps to guide multi-hop question answering.
- **FLARE (Jiang et al., 2023):** An active retrieval strategy that triggers information seeking only when the model generates low-confidence tokens.
- **Context Optimization Methods:** Including **Selective-Context (Li, 2023)**, **LongLLMLingua (Jiang et al., 2024)**, and **RECOMP (Xu et al., 2023)**, which focus on compressing and selecting context to optimize information flow to the generator.
- **Search-o1 (Li et al., 2025a):** A proprietary baseline utilizing the OpenAI o1-preview model equipped with search tools, representing state-of-the-art inference-time reasoning.

Learning-Based Methods. These involve training the agent or retriever to enhance performance:

- **DeepRAG (Guan et al., 2025):** Models retrieval-augmented reasoning as a Markov Decision Process (MDP) for adaptive retrieval.
- **IKEA (Huang et al., 2025):** A reinforced agent designed to synergize internal parametric knowledge with external search, optimizing for efficiency.
- **ReasonRAG (Zhang et al., 2025):** Utilizes process-supervised reinforcement learning with fine-grained rewards for query and answer generation.

[3]https://huggingface.co/Qwen/Qwen3-30B-A3B-Thinking-2507

[4]https://huggingface.co/Qwen/Qwen3-4B-Thinking-2507

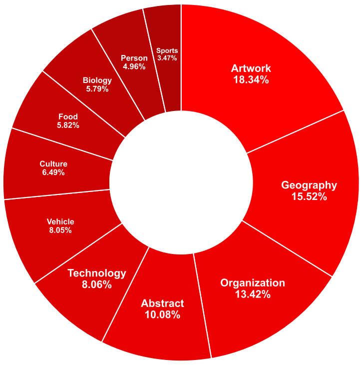

Figure 6: Domain distribution.

- **DecEx-RAG (Leng et al., 2025):** Enhances agentic RAG via decision and execution optimization using process supervision.
- **Search-R1 (Jin et al., 2025):** Utilizes the DeepSeek-R1 model equipped with search capabilities, serving as a representative strong open-weights reasoning model.
- **HL-Data:** A supervised baseline fine-tuned on high-quality human-labeled datasets (combining HotpotQA and 2WikiMultiHopQA) (Yang et al., 2018; Ho et al., 2020). This matches the scale of our synthesized data to serve as a control for data quality.

D Deployment and Inference Details

We deployed gpt-oss-120b[5] and our trained models using the vLLM inference engine on 8×H20 GPUs. For gpt-oss-120b, we set the maximum context length to $100,000$ tokens. The tool-call parser was configured to use the `openai` format. For our trained models, we utilized the `hermes` tool-call parser. All models were served via OpenAI-compatible APIs to maintain a consistent interface. We used FAISS (Johnson et al., 2019) to support fast similarity search.

E Domain Diversity Analysis

To verify the semantic coverage and generalization potential of our synthesized corpus, we conducted a domain analysis of the generated tasks. We employed an LLM to automatically classify the topic of each question-answer pair within the dataset.

RAGShaper achieves exceptional and balanced domain diversity. The resulting distribution, illustrated in Figure 6, demonstrates that RAGShaper achieves exceptional diversity. The synthesized tasks span a broad spectrum of knowledge areas, ranging from humanities and social sciences (e.g., *Artwork*, *Culture*, *Organization*) to STEM fields (e.g., *Technology*, *Biology*). Notably, the distribution is remarkably balanced; no single domain dominates the corpus, with the largest category (*Artwork*) accounting for only 18.34% and substantial representation even in complex categories like *Abstract* (10.08%). This structural uniformity ensures that the trained agent avoids overfitting to specific entity types and instead acquires generalized retrieval and reasoning skills applicable across diverse real-world contexts.

[5] https://huggingface.co/openai/gpt-oss-120b

Trajectory Example: Handling Doppelgänger Entities

User Question: A major literary work commissioned by the Holy Roman Emperor whose reign began in 1508 was part of his grand artistic legacy. While this patron commissioned famous manuscript anthologies during this period, this specific allegorical epic was distinctively designed for the printing press to ensure a wider audience. What is the exact publication year of its first edition?

[Thought 1] I need to identify the work and its date based on the constraints... I will search for works commissioned by Maximilian I during this period...

[Action 1] `Dense_Search(query="Maximilian I commissioned literary works completed 1508-1519 manuscript vs print")`

[Observation 1]

Candidate A (Source 1):
"The **Ambraser Heldenbuch**... written by Hans Ried from **1504 to 1516**. It is a splendid **manuscript**..."
[Analysis: Potential Doppelgänger] Completion (1516) fits the reign. However, the source identifies it explicitly as a "manuscript," which conflicts with the print constraint.

Candidate B (Source 2):
"The allegorical epic **Theuerdank**... The first edition was **printed** in Nuremberg and **published in 1517**..."
[Analysis] Publication (1517) fits the reign. The format (printed) matches the constraint.

[Thought 2] The retrieved documents mention two major works associated with Maximilian I during this period: the *Ambraser Heldenbuch* and *Theuerdank*.

I need to strictly apply the format constraint. The snippet regarding the *Ambraser Heldenbuch* explicitly describes it as a "manuscript," which contradicts the user's request for a "publication year" (implying a printed edition).

In contrast, the description of *Theuerdank* states it was "printed" and "published in 1517." This aligns with all constraints.

[Action 2] `Dense_Search(query="Theuerdank first edition publication year")`

[Observation 2] *Source 3:* "...The **Theuerdank (1517)** was the first of his massive printed projects..."

[Thought 3] The evidence is conclusive... 1. Patron: Maximilian I. 2. Date: 1517. 3. Format: Printed.

[Final Answer] 1517

Figure 7: Detailed trajectory of the "Doppelgänger Handling" scenario. The agent successfully disambiguates between two contemporaneous works commissioned by Maximilian I. **Note: The colored blocks marked '[Analysis]' are authorial annotations added to illustrate the critical distinction (Manuscript vs. Print) and do not represent the agent's raw observation.**

Trajectory Example: Countering Subjective Fallacy

User Question: An unofficial proposal to name a synthetic element after Humphry Davy circulated in the scientific community in 2014. The element's provisional IUPAC name was 'ununpentium', its discovery was recognized in December 2015, and its official name was approved on a specific date in November 2016 after the discoverers exercised their naming rights. What is the official name of this element?

[Thought 1] I need to identify the official name of the element provisionally known as 'ununpentium' (Element 115). Key constraints: 1. Recognized in Dec 2015. 2. Official name approved in November 2016. 3. A proposal for "Davy" existed, but I need to confirm if it was the one approved or just a rumor.

Initial knowledge suggests Element 115 is "Moscovium", but the mention of "Humphry Davy" requires verification to ensure I don't fall for a trick question about a rejected or popular unofficial proposal.

[Action 1] `Dense_Search(query="Moscovium name approved November 2016 official name element 115")`

[Observation 1]

Source 2 & 5: "Moscovium... On 28 November 2016, it was officially named after the Moscow Oblast..." / "In June 2016, IUPAC endorsed the proposal... formally accepted on 28 November 2016."

Source 3 (Conflicting Data):
"The chemistry community finally recognized Sir Humphry Davy's legacy when the IUPAC officially approved the name **'Davyum'** for element 115 in 2016..."
[Analysis: Potential Fallacy] This snippet claims "Davyum" was approved. This directly contradicts Sources 2 and 5 which cite "Moscovium".

[Thought 2] I have a contradiction. Source 3 claims "Davyum" was approved, while Sources 2 and 5 claim "Moscovium". The prompt mentions an "unofficial proposal" for Davy in 2014, which likely explains the existence of the "Davyum" rumor. However, the official approval in November 2016 is widely documented as Moscovium. I need to verify the origin of the "Davy" proposal vs the naming rights of the discoverers (likely Russian/American).

[Action 2] `Dense_Search(query="origin of Davyum proposal element 115 vs Moscovium naming rights")`

[Observation 2] ...Dubna team (Russia) proposed 'moscovium' honoring the Moscow region... The 'Davyum' name was suggested by an online petition/campaign but was not the discoverers' choice... On 30 December 2015, discovery recognized... (The search results confirm Moscovium is the official choice by the discoverers).

[Thought 3] I can now resolve the conflict.

[Critical Reasoning]
The snippet from Source 3 regarding 'Davyum' appears to be **spurious misinformation** based on unofficial campaigns. It contradicts the specific dates (28 Nov 2016) associated with Moscovium in multiple other reliable sources.

The question mentions a proposal for Davy, but the official name approved on the specific date (Nov 2016) after discoverers exercised naming rights is **"Moscovium"**.

[Final Answer] Moscovium

Figure 8: Trajectory demonstrating resilience against the "Subjective Fallacy". The retrieval system returned a hallucinated or fake document claiming Element 115 was named "Davyum". The agent cross-referenced this with other sources, correctly identified the distractor as misinformation derived from unofficial proposals, and reasoned using the logic highlighted in the green block, thereby demonstrating the system's **robustness against conflicting information** and hallucinated sources.

Question: The Filipino statesman who established the government-in-exile during the outbreak of World War II was also the mayor of what city?

Thought:
I need to identify a Filipino statesman who established a government-in-exile during WWII and also served as mayor of a city. I'll start by searching for Filipino statesmen who established government-in-exile during World War II and were also mayors.

```
Query:
"Filipino statesman who established the government-in-exile during World War II also mayor of which
city"
Observation:
"Tomas Confesor... was the governor of Iloilo and later all of Panay Island during the Japanese
Occupation... Right afte...
```

Thought:
The results mention Manuel L. Quezon in connection with the government-in-exile, but don't clearly show him as a mayor. I need to specifically search for Manuel L. Quezon's mayoral role.

```
Query:
"Manuel L. Quezon mayor of city"
Observation:
"Quezon City... President Manuel L. Quezon acted as mayor from October 12 to November 4, 1939,
pending the resignation f...
```

Thought:
The results show Quezon acted as mayor, but I need to confirm which city. The context suggests it's Quezon City. Let me search specifically for Quezon City mayor in October 1939.

```
Query:
"Quezon City mayor October 12 1939"
Observation:
"Quezon City... President Manuel L. Quezon acted as mayor from October 12 to November 4, 1939,
pending the resignation f...
```

Thought:
I've confirmed Quezon was mayor of Quezon City. Now I need to verify that he established the government-in-exile during WWII to ensure I have the complete answer.

```
Query:
"Manuel L. Quezon government-in-exile World War II"
Observation:
"Manuel L. Quezon... He established a government-in-exile in the U.S. with the outbreak of the war
and the threat of Jap...
```

Thought:
I have confirmed that Manuel L. Quezon established the government-in-exile during WWII and served as mayor of Quezon City from October 12 to November 4, 1939. The answer is Quezon City.
<RAG_FINAL_ANSWER>Quezon City</RAG_FINAL_ANSWER>

Figure 9: Trajectory generated by 4.5k RAGShaper trained on the Qwen3-30B-A3B-Thinking-2507. The figure shows the agent solving a multi-hop question: it first identifies the likely candidate (Manuel L. Quezon) from the "government-in-exile" clue, pivots to verify the specific mayoral role (Action 2), discovering he acted as mayor of **Quezon City**, and finally cross-references the WWII context (Action 3) to validate the answer.

Tool Schema

Dense Search


```
{
    "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"]
        }
    }
}
```


Figure 10: Tool schema definition for the dense vector retrieval tool (*query_knowledge_base_dense*).

Core Prompt of Exploration in Information Curation

=== PRIMARY GOAL === Sample a trajectory that will later support a **LOW-ENTRANCE but DEEP multi-hop QA**. You are not just collecting facts — you are building a dependency chain (A→B→C→D...) plus confusable-but-disambiguatable negative documents.

=== SAMPLING STRATEGY & RULES ===
1) Build a Multi-hop Backbone (Depth-First Chain)
- Target $\geq$ 10 dependent hops whenever possible (A→B→C→D...).
- Each retrieval step MUST unlock a **NEW** entity/relation needed for the NEXT hop.
- Do NOT get stuck circling the same entity. Revisit only to cross-check hard metadata.

2) Pack Compact Evidence per Hop
- Capture 1–2 short, quotable snippets per hop that clearly state the relation.
- Capture at least ONE **hard metadata item** (year/date/version/ID/count) that can be cross-checked.
- Ensure FINAL answer-critical metadata is supported by $\geq$ 2 independent observations.

3) Generate Negative Docs Early & Repeatedly
- **Tool Usage:** You must use `write_distractor_docs` (pass `distractor_texts` list). Do NOT call an LLM for this; write the text yourself.
- **Timing:** Create negative docs after the FIRST successful retrieval and after each key hop (especially when hard metadata appears).
- **Quantity:** Min $\geq$ 3 calls total; total $\geq$ 5 distractor documents per seed. Diversify dimensions.

4) Safety Rule: Disambiguation is Mandatory
- The solver cannot know which doc is a distractor. Every negative doc MUST be logically distinguishable (e.g., specific year, version, or scope).
- *Bad:* "Founded in 2015" vs "Founded in 2016" with no other context.
- *Good:* "2015 Annual Report (Audited)" vs "2016 Preliminary Draft".

=== DIMENSION GUIDANCE (Types of Negative Docs) ===
- **[A] Doppelganger:** Adjacent-edition doc (e.g., 2015 vs 2016 manual). Change one spec/value but keep the rest similar. Make the edition explicit.
- **[B] False Shortcut:** A doc claiming A→C directly (skipping B) with hedged phrasing, contradicting the true A→B→C chain.
- **[C] Fragmented Puzzle:** Docs containing only a subset of information, looking locally plausible but incomplete.
- **[D] Subjective Fallacy:** Review/Opinion tone with one plausible factual objective error (e.g., wrong model number).

Figure 11: The **Core Exploration Trajectory** prompt. Unlike standard retrieval, this prompt drives the agent to proactively construct deep dependency chains (10+ hops) and synthesize "Doppelgänger" or "False Shortcut" negative documents during the rollout, laying the groundwork for high-complexity puzzle generation.

Prompt: Trajectory-to-QA Synthesis

Please synthesize a high-quality Q&A pair based on the trajectory:

Question Requirements (Crucial for Reasoning & Brevity): - The target answer must be a specific fact (e.g., a name, a date, a location, a count, or a yes/no status).
- **DO NOT** ask "How", "Why", or "Describe" questions that require long textual explanations.
- **Anti-shortcut**: The question MUST NOT contain the answer text, and MUST NOT directly state the asked attribute in a definitional clause.
- **Low-entrance, deep-reasoning**: Keep the question to <=2 sentences and a small number of top-level clues; depth should come from a multi-hop dependency chain, not a long list of trivia.
- **Deep multi-hop (required)**: The question must require >=3 dependent hops to solve (chain dependency only; no star-shaped checklist).
- **Negative-doc confusability (required)**: If the trajectory includes negative docs (e.g., generated via write_distractor_docs), craft the question so that a careless solver could be misled by at least one negative doc into a plausible wrong answer/path, while the correct answer is still supported by authoritative evidence in the trajectory. - The question should be a natural, factual, and self-contained question (e.g., don't include "What did the agent find...", "what is in the trajectory...", "according to the trajectory...", ...). It must seem like it never undergos a trajectory exploration in previous step. And don't mention "search" or "search results", or things like them.

Answer Requirements (Crucial for Strict Length): - **Extreme Brevity**: The answer MUST be **less than or equal to one sentence, and contain only one entity**, or ideally just a **short phrase** (e.g., "1985", "The Treaty of Versailles", "Increased by 5%").
- **No Fluff**: Do not use filler words like "According to the documents..." or "The answer is...". Provide ONLY the final answer value.
- **Groundedness**: The specific fact must be strictly derived from the provided trajectory observations without mentioning the trajectory or observation.

Required Explanations (for dataset traceability; NOT part of the question text):
- reasoning_steps: Provide >= 3 short, dependent steps that solve the QA using ONLY the trajectory evidence.
- negative_aspect: Explain how negative doc(s) could mislead and what disambiguation defeats them. Mention the distractor dimension when possible.
- disambiguation: How to disambiguate the misleading claim.
- distraction_text: The text that is used to distract the solver.
Return JSON EXACTLY in this schema (do not add extra fields):

```
{
  "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"}
  ]
}
```

Figure 12: The **QA Synthesis** prompt. This prompt consumes the trajectory generated in the previous step. It enforces strict constraints to ensure the synthesized question is "low-entrance" (concise) yet "deep-reasoning" (requires traversing the full dependency chain), and explicitly validates the effectiveness of the negative documents.

Prompt for Trajectory Rollout

You are a helpful assistant. You need to use tools to solve the problem. You have access to a Dense Retrieval system (semantic/vector search). You MUST use the dense retrieval tool to answer and verify.

Core Capabilities
- **Semantic Understanding**: The system matches the *meaning* of your query, not just exact words.
- **Handling Paraphrasing**: It can find relevant content even if different terminology is used.

Query Formulation Strategy
1. **Be Descriptive**: Write natural language queries that fully describe what you are looking for. - *Bad*: "revenue 2023" - *Good*: "What was the total revenue of the company in the fiscal year 2023?"
2. **Context Matters**: Include necessary context in the query string, as the retriever processes independent queries.
3. **Iterative Refinement**: - If results are too broad: Add specific constraints to your query. - If results are irrelevant: Rephrase the query using synonyms or related concepts.

Execution Protocol
1. Break complex multi-hop questions into separate, simpler queries.
2. Verify the retrieved content matches the user's intent.
3. If after multiple attempts (>5) no relevant information is found, try rephrasing your queries with different approaches.

Internal Knowledge Fallback Mechanism
When you have attempted multiple retrieval queries over several rounds but still cannot find the answer in the knowledge base, you should use your internal knowledge to provide the best possible answer. This is a fallback mechanism to ensure you can still help the user even when the knowledge base doesn't contain the required information. When using internal knowledge, clearly indicate this in your reasoning and wrap your answer in the final answer tags.

Critical Requirements
1. **Reasoning-Tool Consistency**: If your reasoning mentions using a tool (e.g., "Let's search", "We need to use the dense retrieval tool"), you MUST generate the corresponding tool_calls. Do not stop at reasoning alone.
2. **Action Follow-through**: If you decide to use a tool in your reasoning, you must follow through with the actual tool call. Empty content with reasoning about tool usage is NOT a valid final answer.

Answer Strategy
1. The final answer should only contain the short answer to the question (few words), avoiding unnecessary reasoning content in the final output string.
2. **MANDATORY**: You MUST wrap the final answer inside {FINAL_ANSWER_START} and {FINAL_ANSWER_END} tags. Never provide an answer without these tags. Every response that contains an answer must use these tags.
3. **Answer Quality Requirements**: - The answer must be a specific entity: a name, place, number, date, ID, or other concrete information.
- **DO NOT** use common words like "and", "or", "the", "of", "in", "is", "was", "are", "were", "a", "an", "as", "for", "with", "from", "to", "on", "at", "by", "this", "that", "these", "those" as your final answer.
- Common words, articles, prepositions, and conjunctions are NOT valid answers. The answer should be a meaningful entity or piece of information that directly answers the question.
- If the retrieved information does not contain a clear answer, indicate that you cannot find the answer, but still wrap your response in the answer tags. 4. Keep any reasoning or explanation outside the {FINAL_ANSWER_START} and {FINAL_ANSWER_END} tags.

Figure 13: The full prompt used during the **trajectory rollout** phase to guide the agent in generating training data. It explicitly instructs the model on query formulation strategies, fallback mechanisms, and the strict formatting required for the final answer.

Prompt for Evaluation

You are a helpful assistant. You need to use tools to solve the problem. You have access to a Dense Retrieval system (semantic/vector search). You MUST use the dense retrieval tool to answer and verify. Do not attempt to use sparse retrieval tools as they are not available.

Core Capabilities
- **Semantic Understanding**: The system matches the *meaning* of your query, not just exact words.
- **Handling Paraphrasing**: It can find relevant content even if different terminology is used.

Query Formulation Strategy
1. **Be Descriptive**: Write natural language queries that fully describe what you are looking for. - *Bad*: "revenue 2023" - *Good*: "What was the total revenue of the company in the fiscal year 2023?"
2. **Context Matters**: Include necessary context in the query string, as the retriever processes independent queries.
3. **Iterative Refinement**: - If results are too broad: Add specific constraints to your query. - If results are irrelevant: Rephrase the query using synonyms or related concepts.

Execution Protocol
1. Break complex multi-hop questions into separate, simpler queries.
2. Verify the retrieved content matches the user's intent.
3. If after multiple attempts (> 5) no relevant information is found, admit that the information is missing from the knowledge base.

Answer Strategy
1. The final answer should only contain the short answer to the question (few words), avoiding unnecessary reasoning content in the final output string.
2. Wrap the final answer inside <RAG_FINAL_ANSWER> and </RAG_FINAL_ANSWER>, and keep any reasoning outside the tokens.

Available Tools
- query_knowledge_base_dense: [Dense Search] Semantic vector search over the knowledge base. Falls back to configured top_k or 5.

Figure 14: The prompt utilized during the evaluation phase. Compared to the training prompt, this version instructs the model to prioritize honesty by admitting when information is missing from the knowledge base, rather than falling back to internal knowledge. It also specifies XML-style tags for the final answer extraction.