

# WEBSHAPER: AGENTICALLY DATA SYNTHESIZING VIA INFORMATION-SEEKING FORMALIZATION

000  
001  
002  
003  
004  
005 **Anonymous authors**  
006 Paper under double-blind review  
007  
008  
009  
010

## ABSTRACT

011 The advent of Large Language Model (LLM)-powered agents has revolutionized  
012 artificial intelligence by enabling solutions to complex, open-ended tasks through  
013 web-based information-seeking (IS) capabilities. The scarcity of high-quality  
014 training data has limited the development of IS agents. Existing data synthesis  
015 approaches typically adopt an *information-driven* paradigm that first collects in-  
016 formation and then refines question-answer pairs through retrieval. However, this  
017 may lead to inconsistency between information structure and reasoning structure,  
018 as well as between the question and the corresponding answer. To mitigate, we  
019 propose a *formalization-driven* IS data synthesis framework WebShaper, which  
020 systematically formalizes IS tasks using set-theoretic constructs. Central to the  
021 formalization is the concept of Knowledge Projections (KP), which enables precise  
022 control over reasoning structure by KP operation compositions. During synthesis,  
023 we begin by creating seed tasks, then use a multi-step expansion process. At  
024 each step, an agentic Expander expands the current formal question more complex  
025 through retrieval and validation tools grounded in our formalization. We train  
026 our model on the synthesized dataset. Experiment results demonstrate that Web-  
027 Shaper achieves state-of-the-art performance among open-sourced IS agents on  
028 competitive benchmarks.

## 1 INTRODUCTION

029  
030  
031 The emergence of Large Language Model (LLM)-powered language agents has marked a paradigm-  
032 shifting advance in artificial intelligence, enabling transformative solutions to previously intractable  
033 challenges across domains (Guo et al., 2024; Wang et al., 2024; AutoGPT, 2023; Wu et al., 2023; Ye  
034 et al., 2023). Information-seeking (IS) represents a core component of the cognitive autonomy of  
035 language agents. This capability not only underpins their adaptability in open-ended tasks but also  
036 powers a range of powerful commercial systems such as Deep Research of OpenAI (OpenAI, 2025),  
037 Gemini (Gemini, 2025), and Perplexity (Perplexity, 2025).

038 Current agentic systems for unlocking this capability typically follow a well-established pipeline  
039 in agent development: (1) First, construct task-specific trajectories of question-answer pairs; (2)  
040 Employ supervised fine-tuning (SFT) to acquire foundational skills (Sun et al., 2025). (3) Generalize  
041 strategic decision-making through on-policy reinforcement learning (RL) (Jin et al., 2025). The entire  
042 development of the IS agent originates from and its ultimate effectiveness depends on high-quality  
043 IS task training data. However, due to its complexity, such a high-quality dataset is both sparse and  
044 difficult to construct through crowdsourcing. **Thus, constructing training data through a carefully  
045 designed agent pipeline becomes the cornerstone of effective IS agent development.**

046 Existing IS dataset synthesis methods typically involve freely pre-searching for information online and  
047 employing LLMs to generate questions from the collected content. These approaches first organize  
048 the collected information into structured formats, then prompt the LLM with the structured data to  
049 produce natural language (NL) questions. Their core objective is to map *information structures* into  
050 *reasoning structures* within the resulting NL questions. Representative methods like WebDancer (Wu  
051 et al., 2025a) and TaskCraft (Shi et al., 2025) generate linear information chains, while others  
052 construct graphs connected via web links (Wu et al., 2025b) or entity coreference networks (Li  
053 et al., 2025a). However, these information-driven approaches face two critical limitations. **First**,  
the synthesis using LLM may struggle to fully comprehend the information structure, resulting

054 in inconsistent reasoning structures or incorrect answers to the generated NL questions. **Besides**,  
 055 disordered information retrieval will lead to excessive data processing and will collect redundant  
 056 homogeneous information structures, which limits the diversity of information structures.  
 057

058 To overcome these limitations, we propose WebShaper<sup>1</sup>, a formalization-driven IS data synthesis  
 059 paradigm. Unlike prior approaches, WebShaper first formalize information-seeking tasks and then  
 060 systematically guide data synthesis through this formalization. During generation, information  
 061 collection is explicitly controlled by formal task requirements. This framework offers three key  
 062 advantages: **Broader Task Coverage**: Systematic exploration of task formalizations enables synthe-  
 063 sizing diverse information-seeking patterns unconstrained by pre-retrieval content limitations; **Task**  
 064 **Controllability**: Explicit formalization parameters allow precise specification of reasoning structures  
 065 and complexity levels; **Structural and Answer Consistency**: Due to the inherent interpretability and  
 066 verifiability of formalized representations, synthesized outputs exhibit fewer inconsistencies across  
 067 both information-reasoning structures and question-answer pairs. With this structured guidance, we  
 068 produce consistent reasoning and redundancy while ensuring rich, diverse reasoning logic.  
 069

070 At the core of our framework lies a formalization of IS tasks, which enables principled and systematic  
 071 generation of task instances with controllable collection complexity and reasoning structures. Unlike  
 072 relevant fields, where there exists task formalization in advance, such as Lean 4 language (Moura &  
 073 Ullrich, 2021) in math proving and propositional logic in knowledge-centric question answering (Xia  
 074 et al., 2025), there's no established formalization for information-seeking. We treat IS as a unified  
 075 problem space where task is systematically derived from compositions of basic units termed Knowl-  
 076 edge Projections (KP). To align with the formalized structure, we initiate synthesis by constructing  
 077 foundational seed tasks, followed by a multi-step expansion grounded in our formal framework. This  
 078 process employs a dedicated agentic Expander module designed to interpret task requirements via KP  
 079 representations. At each expansion stage, the expander transforms the current formal question into a  
 080 more complicated one. It implements layer-wise expansion mechanisms that minimize redundancy  
 081 while preventing reasoning shortcuts through controlled complexity progression. This process ensures  
 082 a broad coverage of the formalized task space and the correctness of the question and answer.  
 083

084 We conduct extensive experiments to validate WebShaper dataset by training agents. Comparison  
 085 with the existing training dataset shows the effectiveness of WebShaper. WebShaper achieves best  
 086 performances among all open-source IS agents on the GAIA and WebWalkerQA benchmarks. Further  
 087 discussions demonstrate the validity of each module of our method. We summarize contributions:  
 088

- 089 • We introduce WebShaper, a formalization-driven data synthesis method for information-seeking  
 090 agents, grounded in our proposed task formalization. Leveraging this method, we construct the  
 091 WebShaper dataset, which enables systematic generation of IS instances.
- 092 • We propose an agentic Expander that iteratively generates and validates questions in alignment  
 093 with the formalization, yielding broad coverage IS training data.
- 094 • We conduct extensive experiments across multiple benchmarks to evaluate WebShaper. Empirical  
 095 results demonstrate that models trained with WebShaper consistently outperform baselines,  
 096 confirming the value of our formalization and synthesis approach.

## 097 2 INFORMATION-SEEKING FORMALIZATION

098 In this section, we introduce our formalization of the information-seeking task. We illustrate an  
 099 example in Figure 1. An information-seeking task  $q(T)$  aims to search for knowledge and facts  
 100 prompted by given facts and locate the answer entity set  $T$ . For a basic example also shown in  
 101 Figure 1:  $q(T) = \text{Which player of a team in the 2004-05 season, who was born in 90s? This team is}$   
 $\text{founded in 1966 and is an East German football team.}$  To solve it, one should seek information about  
 $\text{This team is founded in 1966 and is an East German football team}$  to find that the team is *Berliner*  
 $\text{FC Dynamo.}$  And then seek for players of *Berliner FC Dynamo* team in 2004 and 2005 respectively  
 $\text{and players born in 90s, then reason the answer } T = \{ \text{Robert Rudwaleit, Danny Kukulies, } \dots \}.$

102 Let  $\mathcal{E}$  denote the universal set of entities (e.g., players, teams, years). Let  $R \subseteq \mathcal{E} \times \mathcal{E}$  denote a  
 103 subspace of entity pairs where they have a certain relation. For example, if the relation is *bornIn*,  $R$   
 104 stands for all pairs of (*person*, *year*) where *person* is born in *year*.

105 For a subset  $V \subseteq \mathcal{E}$  and a sub-space  $R$ , define a Knowledge Projection (KP):

$$106 \quad R(V) = \{u \mid \exists v \in V, (u, v) \in R \text{ or } (v, u) \in R\}. \quad (1)$$

107 <sup>1</sup>Without loss of generality, we use WebShaper to denote our data method, dataset, and model.

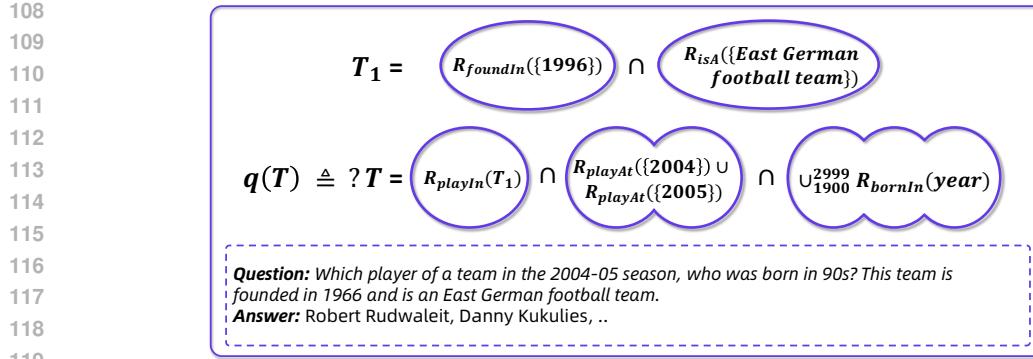


Figure 1: A question-answer case in our information-seeking formalization. We use the **purple** diagram to represent a knowledge projection, which is a set of entities.

For example, when  $R$  denotes entity pairs of relation *bornIn*,  $R(\{90s\})$  represents the set of all people born in 90s. **A KP is the set of entities under a certain relation to other entities, which is the basic unit in an information-seeking task.** KP has two operations:

**$R$ -Union  $\cup$**  In IS, the question may be seeking for a broader condition due to uncertainty about the target. For instance, we only know the target player was playing between 2000-2010 rather than the exact year in advance. The condition can not be more specific than a year range.

Therefore, given  $S_1, S_2$  be entity sets and  $R$ , then:

$$R(V) = R(S_1) \cup R(S_2) \cup \dots \cup R(S_m) \quad (2)$$

represents  $R(V)$  is the union result set in which the entities have a certain relation to entries in either  $S_1, S_2, \dots, S_m$ . If  $R$  stands for relation *playAt*, then the set of players who play between 2000-2010 is  $R(\{2000\}) \cup R(\{2001\}) \cup \dots \cup R(\{2010\})$ .

**Intersection  $\cap$**  Some IS tasks require the target to satisfy several conditions simultaneously. It's interpreted as an Intersection operation of KP:

$$R(V) = R_1(S_1) \cap R_2(S_2) \cap \dots \cap R_n(S_n) \quad (3)$$

where  $R_i$  are about different relations. For example, if  $R_1$  is about *playAt* and  $R_2$  is about *bornIn*, then  $R_1(\{2000\}) \cap R_2(\{90s\})$  stands for players playing in 2000 and born in 90s.

Based on  $R$ -Union and Intersection operations, we define  $T$  as a target set:

$$T = \bigcap_{i=1}^p (R_i(S_{i,1}) \cup R_i(S_{i,2}) \cup \dots \cup R_i(S_{i,t_i})). \quad (4)$$

$S_{i,j} \subset \mathcal{E}$ . More generally,  $T$  can be recursively derived by replacing  $S_{i,j}$  with other target set as:

$$T = R_1(T_1) \cap R_2(T_2) \cap \dots \cap R_k(T_k) \quad (5)$$

An IS task  $q(T) \triangleq ?T$  is to find what entities a questioned  $T$  contains. Therefore, the question example can be formalized as shown in Figure 1.

### 3 DATA SYNTHESIS

In this section, we describe the process of our data synthesis with our task formalization. As Eq. (4) shows, an IS task is recursively composed by knowledge projections. In order to better fit the IS task formalization, we start with constructing a seed task, followed by a multi-step expansion approach. This expansion process is built upon our formalization. We then introduce an agentic Expander. It can understand the task formalization with our KP representation. At each expansion step, we implement the layer-wise expansion to reduce redundancy and reasoning shortcuts. The Expander autonomously retrieves knowledge from the internet, constructs and validates the new FPs to obtain the new question. We elaborate on this process in the following sections.

#### 3.1 SEED QUESTION CONSTRUCTION

The first stage is acquiring a substantial volume of diverse and non-trivial seed questions. To enhance acquisition efficiency, we constructed an offline Wikipedia database by downloading all URLs

162 corresponding to Wikipedia articles while preserving the hyperlinks between them. Subsequently, we  
 163 perform random walks across these articles through their preserved connections. By aggregating the  
 164 content from articles traversed during these random walks, we utilize an LLM to generate synthetic  
 165 data instances. Critically, the generated question-answer pairs must be entirely grounded in the  
 166 content from the collected articles, without relying on external knowledge sources.

167 However, the resulting seed questions could be noisy and contain hallucinations. We launch a filtering  
 168 process. We complete all the seed questions by WebDancer framework (Wu et al., 2025a) based  
 169 on the QwQ model (Team, 2025). We perform 5 times rollouts for each question and keep the data  
 170 where there must be at least one rollout correctly answering the question. We finally construct 18k  
 171 seed questions. We denote the harvested seed question as  $q^1(T)$ .

### 172 3.2 AGENTIC EXPANSION

173 Subsequently, we progressively expand seed questions into increasingly complex ones through  
 174  $l$ -step expansion  $q^{l+1}(T) = \text{Expand}(q^l(T))$  guided by the task formalization. However, the IS  
 175 formalization in Eq. (4) is complicated. The nature of recursion and the composition of multiple  
 176 operations are hard to understand. Besides, since the synthesis relies on retrieving new knowledge  
 177 online, there are several intermediate processes, such as knowledge filtering and selection. Therefore,  
 178 we establish an Agentic Expansion. The core of the expansion is the Expander, which is an agent itself  
 179 to autonomously retrieve information and validate the generation. We introduce the KP representation  
 180 for the Expander to understand our IS formalization. Then, we propose the Layer-wise Expansion  
 181 Strategy to mitigate the limitations of redundant and reasoning shortcuts.

#### 182 3.2.1 KP REPRESENTATION

183 Since  $q(T)$  contains recursion and composition of  $R$ -Union and Intersection operations, it's not trivial  
 184 to represent  $q(T)$  in the Expander agent prompt. We introduce our KP Representation. The key to  
 185 this representation is to: 1) represent a KP unit. 2) can handle  $R$ -Union and Intersection operations.  
 186 3) can handle recursions of KPs. We introduce Constant and Variable. A constant is a subset of  
 187  $\mathcal{E}$  explicitly defined by its elements, e.g.,  $\{90s\}, \{2004, 2005\}$ . A variable is a subset of  $\mathcal{E}$  whose  
 188 elements are not explicitly given. It may appear as a symbolic placeholder in an expression.

189 We use a triplet  $[X, r, S]$  to represent a KP  $R(S)$ .  $r$  is the name of the relation  $R$ .  $X$  is a variable  
 190 while  $S$  can be a variable or a constant. We use the prefix  $V@$  followed by a variable to denote the  
 191 variable  $V$ . We use the prefix  $@C$  before its natural language description to represent a constant. For  
 192 example,  $R_{\text{bornIn}}(\{90s\})$  is represented as  $[@V, \text{bornIn}, 90s]$ . The Intersection operation in Eq.(3)  
 193 can be naturally represented as a list of triplets  $[[X, r_1, S_1], [X, r_2, S_2], \dots, [X, r_n, S_n]]$ .

194 For the  $R$ -Union in Eq.(2), simply expressing it in a list-like form will make the representation  
 195 complicated in recursive  $R$ -Union and Intersection. We notice  $R$ -Union has the following proposition:

196 **Proposition 1.** *For a certain  $R$ ,  $R$ -union satisfies the distributive Law:*

$$198 R(S_1) \cup R(S_2) = R(S_1 \cup S_2) \quad (6) \\ 199$$

200 We leave the proof in the Appendix C. With this proposition, we represent the  $R$ -Union of KP  
 201 by a merge set  $S_1 \cup S_2$ . In practice, we express the union of sets by induction (eg.  $\{1990\} \cup$   
 202  $\{1991\} \cup \dots \cup \{1999\}$  as  $\{90s\}$ ). Or simply add underlines between them (eg.  $\{1990\} \cup \{1991\}$ )  
 203 as  $\{1990\_1991\}$ ). After that, our representation would only have an intersection between triplets.

204 By introducing variables, our representation naturally handles KP recursion by flattening it into the  
 205 intersection of KPs. For example, given a recursion  $R^1(R^2(S))$ , we can represent it as  $[[V@X, r_1,$   
 206  $V@Y], [V@Y, r_2, S]]$ . Finally, an IS task  $q(T)$  can be represented by a list of triplets. The example  
 207 question in Figure 1 can be represented as:

$$209 q(T) \triangleq ?T \quad s.t. \quad [[V@T, \text{playIn}, V@X], \quad [V@T, \text{playAt}, C@2004\_05], \\ 210 \quad [V@T, \text{bornIn}, C@90s], \quad [V@X, \text{foundIn}, C@1966], \quad (7) \\ 211 \quad [V@X, \text{isA}, C@East German football team]]$$

#### 213 3.2.2 LAYER-WISE EXPANSION STRATEGY

214 After representing  $q(T)$ , we outline the iteration-based expansion process. Compared to prior methods  
 215 extending questions in natural language, our IS task formalism enables systematic structural analysis,  
 revealing latent patterns and enabling controlled, optimized expansion. To clearly illustrate the

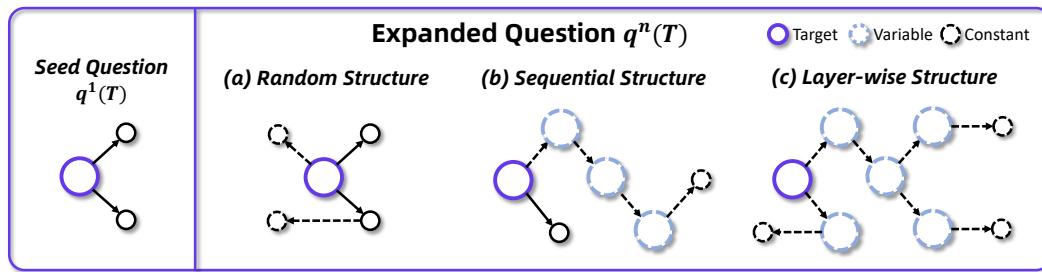


Figure 2: (a) **Random Structure** denotes expanding by randomly adding constants. (b) **Sequential Structure** is expanding on a chain of reasoning sequence. (c) **Layer-wise Structure** traverses layer-wisely on leaf constants and replaces them with variables. “Target” stands for target variable.

expansion strategy, we show our KP representation in a graph: nodes are variables/constants from triplets; edges are relations. For example, Eq. (7) maps to the seed question in Figure 2, where the target variable is determined via given constants. Earlier approaches lacked formal structure, producing either **Random** (Wu et al., 2025b; Shi et al., 2025) (adding FP to arbitrary nodes, Figure 2(a) or **Sequential** (Wu et al., 2025a) (linear reasoning chains, Figure 2(b), both with issues:

- *Redundancy*: Constants directly linked to other constants add sentences like “Dynamo Berlin is a football club based in Berlin” without extending reasoning.
- *Reasoning Shortcut*: Constants that are close to or directly connected to the target result in skipping deeper reasoning for the agent.

We address this via Layer-wise Expansion as illustrated in the Figure 2(c). We layer-wisely traverse the graph to find all leaf constants. When we obtain all the leaf constants of the current graph, an Expander takes each constant once to construct this constant into new FPs. These FPs can form a sub-question that regards the constant as the answer. The expander then merges the sub-questions with the current one to form a new one. It replaces those constants with the sub-questions. Note that the answer for the expanded question always remains. The resulting structure would not have the Redundant and Reasoning Shortcut problems. The number of expanding layers  $l$  is a hyperparameter for controlling the task coverage and difficulty.

### 3.2.3 EXPANDER AGENT

We now introduce the Expander, an autonomous agent designed to enhance question generation through iterative refinement. Given an input constant, the Expander first retrieves relevant information, then formulates a semantically coherent sub-question. This sub-question is subsequently integrated with the original query to construct an enriched, context-aware question that better aligns with the underlying information-seeking objective. It builds on ReAct (Yao et al., 2023), cycling through **Thought**–**Action**–**Observation** triples  $(\tau_t, \alpha_t, o_t)$ , where each **Action**  $(\tau, \phi)$  specifies a tool and parameters. We equip the Expander with the following tools:

- **Search**: It enables Expander to conduct Google search by several queries about a constant and obtains search results. The parameters of this tool are  $\phi = \{queries, filter\_year\}$ , enabling temporal filtering of search results. This tool returns top relevant URLs and their snippets.
- **Summarize**: This is the key to *R*-Union operation. This action allows Expander to visit multiple URLs for the constant and summarize the content. The summarization would integrate the retrieved information to obtain a union constant set as stated in Eq.(6). The parameters of this tool are  $\phi = \{urls, goal\}$ . This tool returns the summarization from the given URLs.
- **Validate**: When Expander completes retrieving and summarizing the KPs of constant, it derives a sub-question and uses this tool to validate the results based on our formalization. The validation purposes are to determine: 1) whether the derived sub-question are approximately consistent with the constants based on the formalization. 2) whether it is too simple that can be directly answered by an LLM. This tool would return detailed validation results as **Observation**, and the Expander would take the next action according to it.

### 3.3 TRAJECTORY CONSTRUCTION

We then construct task-solving trajectories using ReAct format. At each step, the agent generates **Thought**, performs **Action**, receives **Observation**, and chooses the next move. At each time step  $t$ , the agent execution loop can be formalized as a triple  $(\tau_t, \alpha_t, o_t)$ , where  $\tau_t$  denotes the free-form

270 Table 1: **Main results** on GAIA and WebWalkerQA benchmarks. We compare WebShaper with several cutting-edge baselines methods. **bolded** number stands for the best results on the corresponding settings. **Blue** scores are the highest among all open-sourced methods.

Backbone	Framework	GAIA				WebWalkerQA			
		Level 1	Level 2	Level 3	Avg.	Easy	Medium	Hard	Avg.
<i>No Agency</i>									
Qwen-2.5-7B	Base	12.8	3.8	0.0	6.8	1.25	0.8	0.7	0.8
Qwen-2.5-32B	Base	20.5	9.6	8.3	13.6	3.8	2.5	3.3	3.1
	RAG	12.8	11.8	8.3	11.8	23.1	14.3	11.3	15.3
Qwen-2.5-72B	Base	20.5	13.5	0.0	14.6	9.4	7.1	3.3	6.3
GPT-4o	Base	23.1	15.4	8.3	17.5	6.7	6.0	4.2	5.5
QwQ-32B	Base	30.8	15.4	25.0	22.3	7.5	2.1	4.6	4.3
	RAG	33.3	36.5	8.3	32.0	36.9	26.1	33.5	31.2
DeepSeek-R1-671B	Base	43.6	26.9	8.3	31.1	5.0	11.8	11.3	10.0
<i>Close-Sourced Agentic Frameworks</i>									
<i>OpenAI DR</i>		74.3	69.1	47.6	67.4	-	-	-	-
<i>Open-sourced Agentic Frameworks</i>									
Qwen-2.5-32B	Search-o1	33.3	25.0	0.0	28.2	-	-	-	-
	WebDancer	46.1	44.2	8.3	40.7	44.3	46.7	29.2	38.4
	<b>WebShaper</b>	61.5	53.8	16.6	<b>52.4</b>	58.1	51.4	47.0	<b>51.4</b>
QwQ-32B	Search-o1	53.8	34.6	16.6	39.8	43.1	35.0	27.1	34.1
	WebThinker-Base	53.8	44.2	16.6	44.7	47.2	41.1	39.2	41.9
	WebThinker-RL	56.4	50.0	16.6	48.5	58.8	44.6	40.4	46.5
	Simple DS	-	-	-	50.5	-	-	-	-
	WebDancer	61.5	50.0	25.0	51.5	52.5	59.6	35.4	47.9
Qwen-2.5-72B	<b>WebShaper</b>	69.2	50.0	16.6	<b>53.3</b>	55.8	49.2	45.4	<b>49.7</b>
	WebSailor	-	-	-	55.4	-	-	-	-
	<b>WebShaper</b>	69.2	63.4	16.6	<b>60.1</b>	56.2	52.1	49.5	<b>52.2</b>

299 **Thought**,  $\alpha_t$  represents the structured **Action**, and  $o_t$  corresponds to the **Observation** returned  
300 by the environment. The **Thought** component  $\tau_t$  is unrestricted natural-language reasoning that the  
301 model uses for planning, decomposition, self-reflection, or grounding intermediate assumptions. The  
302 Action  $\alpha_t$  is further decomposed into an action type  $\alpha^m$  and its parameter set  $\alpha^p$ , i.e.,  $\alpha = (\alpha^m, \alpha^p)$ .  
303 The action type  $\alpha^m \in \{\text{Search, Visit, Answer}\}$  corresponds to the core tool interfaces used in deep  
304 information-seeking tasks<sup>2</sup>.

305 To standardize trajectories and facilitate supervised learning, we adopt explicit structural markers  
306 for each segment. **Thought** segments are enclosed by `<think>` and `</think>`, **Action** segments by  
307 `<tool_call>` and `</tool_call>`, and **Observation** segments by `<tool_response>` and `</tool_response>`.  
308 The final **Action** segment, corresponding to the model’s ultimate response to the task, is encapsulated  
309 in `<answer>` and `</answer>`. These markers make agent behavior transparent and machine-parsable,  
310 enabling precise control, analysis, and dataset construction. Each question gets 5 rollouts. We  
311 remove the trajectories where the answers are wrong, contain hallucinated observations, or severe  
312 repetition. We finally obtain 5,000 trajectories for supervised and reinforcement learning.

### 313 3.4 AGENT TRAINING

314 To train our information-seeking agent, similar to WebDancer (Wu et al., 2025a), we implement  
315 supervised fine-tuning (SFT) followed by reinforcement learning (RL). In SFT, we mask out loss  
316 from observation leading to loss. For RL algorithm, we use GRPO (Shao et al., 2024). We leave the  
317 details in the Appendix F.

## 318 4 EXPERIMENTS

### 320 4.1 EXPERIMENTAL SETUPS

321 We evaluate WebShaper on two information-seeking benchmarks: **GAIA** (Mialon et al., 2023) and  
322 **WebWalkerQA** (Wu et al., 2025b). We use the *LLM-as-Judges* paradigm to evaluate both tasks using

323 <sup>2</sup>The details of tools are shown in App. G

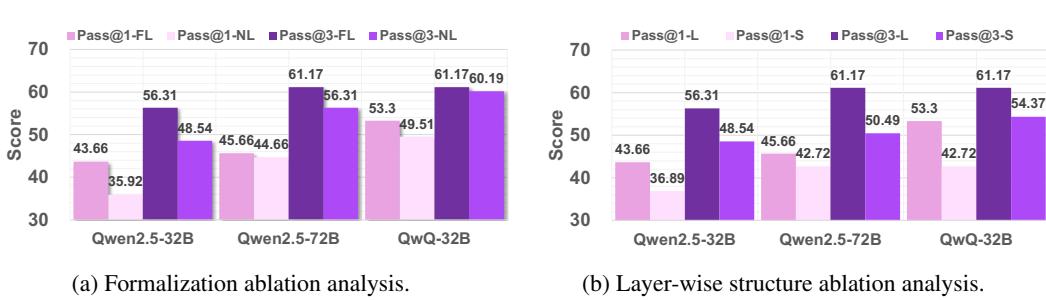


Figure 3: Discussions on formalization and layer-wise structure.

the Pass@1 metric, following [Li et al. \(2025c\)](#). We elaborate on the WebShaper data statistics in the Appendix D. To assess dataset quality, we conduct a comparative study during the SFT stage by training on different data sources. The detailed descriptions of the datasets used in this comparison are provided in Appendix E.

## 4.2 BASELINES

We mainly compare our method to open-source cutting-edge deep research agent frameworks: Search-01 ([Li et al., 2025b](#)), WebDancer ([Wu et al., 2025a](#)), WebThinker ([Li et al., 2025c](#)), SimpleDeepResearch ([Sun et al., 2025](#)), and WebSailor ([Li et al., 2025a](#)). As more strict comparing settings, we train baseline models on WebWalkerQA ([Wu et al., 2025b](#)), E2HQA ([Wu et al., 2025a](#)), and MHQA ([Sun et al., 2025](#)), respectively.

## 4.3 MAIN RESULTS

We compare WebShaper with cutting-edge baselines. The results are shown in Table 1. WebShaper achieves best performances on open-sourced methods on both GAIA and WebWalkerQA. Among all GAIA results, WebShaper-on Qwen-2.5-72B excels second-best method WebSailor 4.7 score. On WebWalkerQA WebShaper obtains the highest 52.2 score. WebShaper performs the best on each backbone setting. These results indicate the generalizability of the synthesized data on different models. WebShaper is currently the only open source method with a score of more than 60 points, which is close to the SOTA OpenAI DR system. WebShaper is implemented fully under open-sourced LLMs, demonstrating that high-quality IS data can deeply stimulate the ability of DR Agents. Notably, we find reinforcement learning on QwQ-32B is not significant. Therefore, we report the SFT result on QwQ-32B in Table 1.

## 4.4 DISCUSSIONS

### 4.4.1 DATA COMPARISON

Table 2: SFT Data Comparison on GAIA benchmarks. The best results are in **bolded**.

Backbone	Dataset	Avg.
Qwen-2.5-32B	WebWalkerQA	32.0
	E2HQA	39.8
	MHQA	35.9
	<b>WebShaper</b>	<b>43.6</b>
Qwen-2.5-72B	WebWalkerQA	38.8
	E2HQA	44.6
	MHQA	43.6
	<b>WebShaper</b>	<b>45.6</b>
QwQ-32B	WebWalkerQA	45.6
	E2HQA	45.6
	MHQA	41.7
	<b>WebShaper</b>	<b>53.3</b>

In this section, we compare WebShaper with baseline datasets. We sample 5,000 data points from each dataset. Then we supervised fine-tune Qwen2.5-32B, Qwen2.5-72B ([Yang et al., 2024](#)), and QwQ ([Team, 2025](#)) on each dataset. The comparative results on GAIA presented in Table 2 demonstrate the superior performance of WebShaper across all backbone architectures on the GAIA benchmarks. Notably, WebShaper achieves the highest average scores for Qwen-2.5-32B, Qwen-2.5-72B, and QwQ-32B, respectively, significantly outperforming baseline datasets like WebWalkerQA and MHQA. Even when comparing models with similar parameter counts (e.g., Qwen-2.5-32B), WebShaper-enabled models show substantial improvements. The consistency of WebShaper’s performance improvement suggests its effectiveness in enhancing model capabilities regardless of architectural design. These findings validate

378 the effectiveness of formalization-driven data synthesis, making it a superior training data solution  
 379 for information-seeking tasks. More details are in the Appendix I.  
 380

#### 381 4.4.2 RL STIMULATION

382  
 383 We compare GAIA performances between mod-  
 384 els trained after SFT and reinforcement learning.  
 385 RL models are trained based on the SFT results.  
 386 As illustrated in Figure 4, our experimental results  
 387 demonstrate significant performance improvements  
 388 across both Qwen2.5-32B and Qwen2.5-72B mod-  
 389 els after RL training on both GAIA and WebWalkerQA.  
 390 The Pass@1 metric shows notable enhance-  
 391 ments of +7.8 points for the 32B model and an  
 392 even more pronounced +13.5 points increase for  
 393 the 72B variant on GAIA. On WebWalkerQA, Web-  
 394 Shaper also improves IS capability on a large scale.  
 395 This substantial gain highlights the critical role of  
 396 RL in activating advanced information-seeking ca-  
 397 pabilities within LLM. The breadth and complex-  
 398 ity of tasks introduced by our task formalization  
 399 stimulate dynamic IS strategies during RL. Unlike  
 400 generic datasets, our carefully curated scenarios  
 401 require the model to iteratively query relevant infor-  
 402 mation, effectively "training" it to prioritize context-  
 403 ually aligned knowledge fragments.  
 404

#### 405 4.4.3 FORMALIZATION

406 In this part, we validate whether our formalization truly improves the dataset. We compare our dataset  
 407 to a variation that uses natural language during the data synthesis. We denote our method with formal  
 408 language as FL, while natural language as NL. This variation takes the current question in each  
 409 iteration and also uses the Expander agent to expand it to a new question. The Expander process in  
 410 natural language as well. We SFT Qwen2.5-32B, Qwen2.5-72B, and QwQ on both datasets. The  
 411 other training setting remains the same. We compare the training results with the variation as shown  
 412 in Figure 3a. FL excels NL in all base model backbones. These results indicate that our formalization  
 413 language can mitigate the limitations incurred by natural language. Our IS task formalization can  
 414 synthesize more forms of tasks. It also reduces error propagation in the synthesis process, leading to  
 415 consistent and precise question-and-answer pairs.  
 416

#### 417 4.4.4 LAYER-WISE EXPANSION STRATEGY

418 We evaluate the effectiveness of the Layer-wise structure. In order to compare, we set up a variation  
 419 which uses the same Expander and task formalization but expands the question in a sequence as  
 420 shown in Figure 2. We SFT Qwen2.5-32B, Qwen2.5-72B, and QwQ on both datasets. Other training  
 421 settings remain the same. We denote method with the layer-wise structure as L, while the sequential  
 422 structure as S. The results as shown in Figure 3b. The layer-wise structure performs better than the  
 423 Sequential structure in all base models. The results show that our method truly mitigates shortcomings  
 424 such as Redundancy and Reasoning shortcuts. Our method improves the final performance via the  
 425 controllable structures.  
 426

#### 427 4.4.5 TOOL CALL ANALYSIS

428 We show the distribution tool call count of the agent to solve a question in different datasets. We  
 429 illustrate the tool call counts larger than 3, which shows the complicated trajectories proportion.  
**Search Complexity (Figure 5a)** WebShaper exhibits a pronounced long-tail distribution. Pretty  
 430 much tasks requiring over 3 search operations. This is 3-4x higher than E2HQA and MHQA,  
 431 indicating superior handling of information-rich queries requiring iterative refinement. **Knowledge  
 Navigation (Figure 5b)** The visit operation distribution shows WebShaper maintains a high ratio for

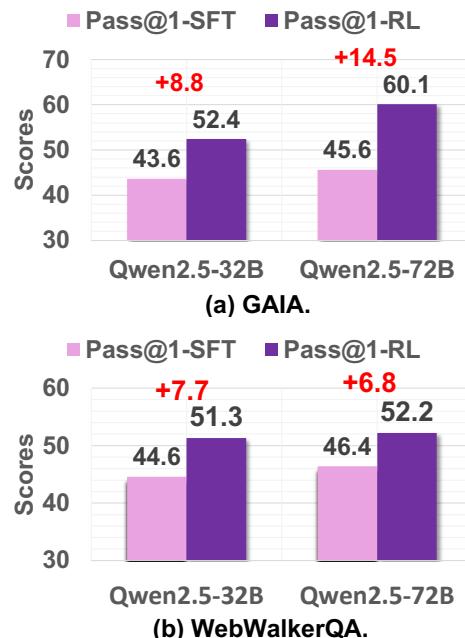


Figure 4: Comparison with SFT and RL.

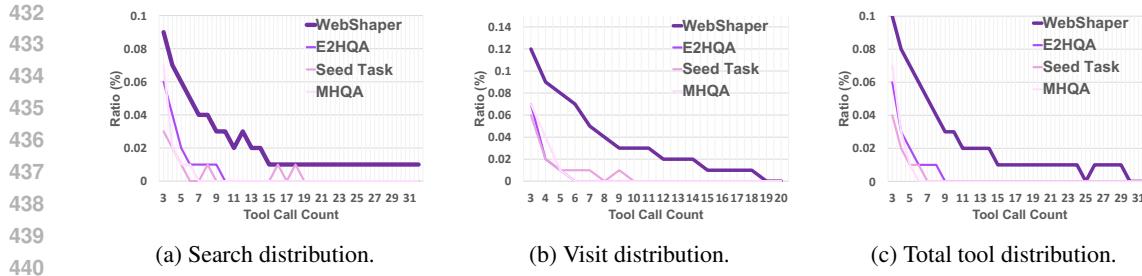


Figure 5: Tool call analysis.

trajectories exceeding 3 steps, while competing datasets sharply drop after 10 steps. This sustained capability reflects enhanced navigational intelligence in IS tasks. **Composite Reasoning (Figure 5c)** In total tool calls, WebShaper’s doubles the count larger than 3. Notably, it sustains non-zero proportions up to 30 tool calls, demonstrating scalability for highly complex compositional reasoning. These findings underscore WebShaper’s unique ability to manage intricate reasoning chains, with statistically significantly higher proportions of multi-hop reasoning trajectories across all modalities. The sustained performance in extended tool call sequences suggests superior architectural capacity for managing complex task decompositions compared to existing benchmarks.

#### 4.5 COST ANALYSIS

Our overall synthesis pipeline is indeed more computationally demanding than traditional information-driven methods. On average, generating a single example requires roughly 20 LLM completions, 6 search calls, 6 visit calls, and about 7 minutes of end-to-end runtime. While approaches such as WebWalkerQA and MHQA incur lower cost by relying on only a few LLM completions, they typically yield simpler multi-hop questions. More advanced methods like E2HQA also rely on multiple LLM and tool calls and are therefore not substantially cheaper when targeting complex reasoning. Importantly, the additional compute is necessary to produce in-domain, high-fidelity GAIA and WebWalker-level data. As shown in Table 2, SFT on our synthesized data improves GAIA performance by 5–10 points across multiple backbones, consistently outperforming WebWalkerQA, E2HQA, and MHQA. These results underscore that the increased computational cost is well justified by the significant gains in data quality and downstream performance.

#### 4.6 QUANTITATIVE EVALUATION OF QA FACTUAL ACCURACY

For the quantitative evaluation of QA factual accuracy, manual verification of large-scale synthesized QA pairs (thousands of instances) is impractical and resource-intensive. We thus adopt a proxy method leveraging the interpretability of our formalization: feeding the formalized structure (including intermediate reasoning steps) of each synthesized question to the QwQ-based solving agent, which provides explicit reasoning path guidance and makes the agent’s answer accuracy a reliable reflection of the original QA pair’s factual correctness. Experimental results show the agent achieves over **80%** accuracy on the synthesized dataset, confirming that our formalization-driven synthesis effectively mitigates hallucinations and inconsistencies while ensuring strong factual correctness.

#### 4.7 ETHIC DISCUSSION

Our synthesis pipeline is specifically engineered to mitigate ethical concerns and data quality risks through three core safeguards. First, seed questions are derived from factual, rigorously curated Wikipedia entries, which effectively mitigates topic drift and minimizes the propagation of inherent biases. Second, the generated tasks are strictly centered on fact-driven reasoning, deliberately steering clear of subjective or sensitive domains, where biases are prone to emerge and amplify. Third, our multi-stage verification process systematically uphold factual consistency, while filtering out hallucinatory content and data contaminated by misinformation or bias.

## 5 RELATED WORK

### 5.1 INFORMATION-SEEKING DATA SYNTHESIS

Recent advances in information-seeking agents aim to integrate web interaction into LLMs’ reasoning. While these works exhibit promising capabilities, they predominantly depend on limited or overly

486 simplistic datasets (Yang et al., 2018; Joshi et al., 2017; Kwiatkowski et al., 2019). Concurrently,  
 487 several recent benchmarks, such as GAIA (Mialon et al., 2023), BrowseComp (Wei et al., 2025), and  
 488 BrowseComp-zh (Zhou et al., 2025), provide only test sets, which restricts their applicability for training  
 489 agents. Early efforts, such as WebWalkerQA, explored simulating human-like web navigation to  
 490 generate QA pairs by constructing linear information chains. CRAWLQA within WebDancer expands  
 491 simple questions to more complex ones by aggregating external information, while SailorFog-QA  
 492 within WebSailor leverages entity coreference networks to support fuzzy reasoning. These methods  
 493 are predominantly information-driven, focusing on strategies for retrieving and connecting knowl-  
 494 edge. In contrast, our approach is formalization-driven, emphasizing the structural representation and  
 495 principled modeling of the QA process.

## 496 5.2 FORMALIZATION-BASED DATA SYNTHESIS

497 Recent work exploits formalization to synthesize training corpora for LLM theorem provers.  
 498 DeepSeek-MathProver translates high-school and undergraduate competition problems into Lean4  
 499 statements, generates proofs with an LLM, and validates them in the Lean4 kernel (Xin et al., 2024).  
 500 DeepSeek-MathProverV2 further decomposes proofs into subgoals and distills subgoal proofs into a  
 501 lightweight model (Ren et al., 2025). Concurrently, Leang et al. (2025) synthesize “prover-as-judge”  
 502 data via iterative natural-language $\leftrightarrow$ formal-language alignment, replacing human feedback in RLHF  
 503 and improving DPO outcomes. Goedel-Prover bootstraps a sequence of successively stronger provers  
 504 on a dynamically expanding Lean4 corpus (Lin et al., 2025). A parallel line applies formalization to  
 505 KBQA. LACT constructs arbitrary first-order logical queries via binary-tree decomposition, yielding  
 506 an SFT dataset that is fine-tuned on an easy-to-hard curriculum (Xia et al., 2025). Departing from  
 507 propositional or FOL formalisms, our work grounds data synthesis in set-theoretic IS.

## 508 6 CONCLUSION

509 This work presents a paradigm-shifting framework for synthesizing training data WebShaper for  
 510 information-seeking (IS) agents through formalization-driven design. By establishing a set theory-  
 511 based mathematical formalization of IS tasks, we address critical limitations in existing information-  
 512 driven approaches that suffer from structural inconsistencies, task controllability, diversity, and  
 513 coverage. The composition of proposed Knowledge Projections enables precise engineering of rea-  
 514 soning structures and complexity. Our agentic Expander module further ensures systematic expansion  
 515 of formalized tasks with a layer-wise expansion paradigm, combining autonomous knowledge re-  
 516 trieval and rigorous validation to minimize redundancy and prevent reasoning shortcuts. Experimental  
 517 results demonstrate that WebShaper not only achieves state-of-the-art performance on GAIA and  
 518 WebWalkerQA benchmarks but also introduces controllability over task design, enabling deliberate  
 519 engineering of cognitive challenges for IS agents. This formalization-driven paradigm shifts the  
 520 focus from reactive information organization to proactive task specification, opening new avenues for  
 521 advancing agent capabilities.

## 522 524 LIMITATION

523 **525 Restricted Tool Set.** Our current framework supports only two tools, search and visit—for infor-  
 526 mation seeking interactions. While this suffices for the benchmarks studied, it limits the agent’s  
 527 ability to handle more complex or multi-step workflows. We plan to incorporate more sophisticated  
 528 tools, including abstracted browser-functionality modules and a Python sandbox environment for  
 529 API-based interactions.

530 **532 Tasks Scope.** Our experiments focus on two short-answer information-seeking benchmarks. How-  
 533 ever, a comprehensive web agent should be capable of tackling broader tasks such as report-level  
 534 research, long-form synthesis, and open-ended generation. Extending our framework to these settings  
 535 is an important direction for future work.

536 **538 Training Context Length Constraint.** The models are trained under a 32k sequence length budget.  
 539 While this is a pragmatic choice, it may restrict the model’s ability to reason over even longer contexts  
 or handle more complex web trajectories. Scaling the context window is a natural next step.

540 REFERENCES  
541

542 AutoGPT. AutoGPT: The heart of the open-source agent ecosystem, 2023. URL <https://github.com/Significant-Gravitas/Auto-GPT>.

543

544 Gemini. Gemini deep research, 2025. URL <https://gemini.google.com/app>.

545

546 Taicheng Guo, Xiuying Chen, Yaqi Wang, Ruidi Chang, Shichao Pei, Nitesh V Chawla, Olaf Wiest,  
547 and Xiangliang Zhang. Large language model based multi-agents: A survey of progress and  
548 challenges. *arXiv preprint arXiv:2402.01680*, 2024.

549

550 Bowen Jin, Hansi Zeng, Zhenrui Yue, Jinsung Yoon, Sercan Arik, Dong Wang, Hamed Zamani, and  
551 Jiawei Han. Search-r1: Training llms to reason and leverage search engines with reinforcement  
552 learning. *arXiv preprint arXiv:2503.09516*, 2025.

553

554 Jina.ai. Jina, 2025. URL <https://jina.ai/>.

555

556 Mandar Joshi, Eunsol Choi, Daniel S Weld, and Luke Zettlemoyer. Triviaqa: A large scale distantly  
557 supervised challenge dataset for reading comprehension. *arXiv preprint arXiv:1705.03551*, 2017.

558

559 Kimi. Kimi deep research, 2025. URL <https://www.kimi.com/>.

560

561 Tom Kwiatkowski, Jennimaria Palomaki, Olivia Redfield, Michael Collins, Ankur Parikh, Chris  
562 Alberti, Danielle Epstein, Illia Polosukhin, Jacob Devlin, Kenton Lee, et al. Natural questions: a  
benchmark for question answering research. *Transactions of the Association for Computational  
Linguistics*, 7:453–466, 2019.

563

564 Joshua Ong Jun Leang, Giwon Hong, Wenda Li, and Shay B Cohen. Theorem prover as a judge for  
565 synthetic data generation. *arXiv preprint arXiv:2502.13137*, 2025.

566

567 Kuan Li, Zhongwang Zhang, Huifeng Yin, Liwen Zhang, Litu Ou, Jialong Wu, Wenbiao Yin, Baixuan  
568 Li, Zhengwei Tao, Xinyu Wang, et al. Websailor: Navigating super-human reasoning for web  
agent. *arXiv preprint arXiv:2507.02592*, 2025a.

569

570 Xiaoxi Li, Guanting Dong, Jiajie Jin, Yuyao Zhang, Yujia Zhou, Yutao Zhu, Peitian Zhang, and  
571 Zhicheng Dou. Search-o1: Agentic search-enhanced large reasoning models. *arXiv preprint  
arXiv:2501.05366*, 2025b.

572

573 Xiaoxi Li, Jiajie Jin, Guanting Dong, Hongjin Qian, Yutao Zhu, Yongkang Wu, Ji-Rong Wen, and  
574 Zhicheng Dou. Webthinker: Empowering large reasoning models with deep research capability.  
575 *CoRR*, abs/2504.21776, 2025c. doi: 10.48550/ARXIV.2504.21776. URL <https://doi.org/10.48550/arXiv.2504.21776>.

576

577 Yong Lin, Shange Tang, Bohan Lyu, Jiayun Wu, Hongzhou Lin, Kaiyu Yang, Jia Li, Mengzhou Xia,  
578 Danqi Chen, Sanjeev Arora, et al. Goedel-prover: A frontier model for open-source automated  
579 theorem proving. *arXiv preprint arXiv:2502.07640*, 2025.

580

581 Grégoire Mialon, Clémentine Fourrier, Thomas Wolf, Yann LeCun, and Thomas Scialom. Gaia:  
582 a benchmark for general ai assistants. In *The Twelfth International Conference on Learning  
Representations*, 2023.

583

584 Leonardo de Moura and Sebastian Ullrich. The lean 4 theorem prover and programming language.  
585 In *International Conference on Automated Deduction*, pp. 625–635. Springer, 2021.

586

587 OpenAI. Deep research system card, 2025. URL <https://cdn.openai.com/deep-research-system-card.pdf>.

588

589 Perplexity. Perplexity deep research, 2025. URL <https://www.perplexity.ai/>.

590

591 ZZ Ren, Zhihong Shao, Junxiao Song, Huajian Xin, Haocheng Wang, Wanja Zhao, Liyue Zhang,  
592 Zhe Fu, Qihao Zhu, Dejian Yang, et al. Deepseek-prover-v2: Advancing formal mathematical  
593 reasoning via reinforcement learning for subgoal decomposition. *arXiv preprint arXiv:2504.21801*,  
2025.

594 Zhihong Shao, Peiyi Wang, Qihao Zhu, Runxin Xu, Junxiao Song, Xiao Bi, Haowei Zhang,  
 595 Mingchuan Zhang, YK Li, Y Wu, et al. Deepseekmath: Pushing the limits of mathematical  
 596 reasoning in open language models. *arXiv preprint arXiv:2402.03300*, 2024.

597

598 Guangming Sheng, Chi Zhang, Zilingfeng Ye, Xibin Wu, Wang Zhang, Ru Zhang, Yanghua Peng,  
 599 Haibin Lin, and Chuan Wu. Hybridflow: A flexible and efficient rlhf framework. In *Proceedings  
 600 of the Twentieth European Conference on Computer Systems*, pp. 1279–1297, 2025.

601 Dingfeng Shi, Jingyi Cao, Qianben Chen, Weichen Sun, Weizhen Li, Hongxuan Lu, Fangchen Dong,  
 602 Tianrui Qin, King Zhu, Minghao Yang, et al. Taskcraft: Automated generation of agentic tasks.  
 603 *arXiv preprint arXiv:2506.10055*, 2025.

604

605 Shuang Sun, Huatong Song, Yuhao Wang, Ruiyang Ren, Jinhao Jiang, Junjie Zhang, Fei Bai, Jia  
 606 Deng, Wayne Xin Zhao, Zheng Liu, et al. Simpledeepsearcher: Deep information seeking via  
 607 web-powered reasoning trajectory synthesis. *arXiv preprint arXiv:2505.16834*, 2025.

608 QwQ Team. Qwq-32b: Embracing the power of reinforcement learning, 2025. URL <https://qwenlm.github.io/blog/qwq-32b/>.

609

610

611 Lei Wang, Chen Ma, Xueyang Feng, Zeyu Zhang, Hao Yang, Jingsen Zhang, Zhiyuan Chen, Jiakai  
 612 Tang, Xu Chen, Yankai Lin, et al. A survey on large language model based autonomous agents.  
 613 *Frontiers of Computer Science*, 18(6):186345, 2024.

614

615 Jason Wei, Zhiqing Sun, Spencer Papay, Scott McKinney, Jeffrey Han, Isa Fulford, Hyung Won  
 616 Chung, Alex Tachard Passos, William Fedus, and Amelia Glaeser. Browsecmp: A simple yet  
 617 challenging benchmark for browsing agents. *arXiv preprint arXiv:2504.12516*, 2025.

618

619 Jialong Wu, Baixuan Li, Runnan Fang, Wenbiao Yin, Liwen Zhang, Zhengwei Tao, Dingchu Zhang,  
 620 Zekun Xi, Yong Jiang, Pengjun Xie, et al. Webdancer: Towards autonomous information seeking  
 621 agency. *arXiv preprint arXiv:2505.22648*, 2025a.

622

623 Jialong Wu, Wenbiao Yin, Yong Jiang, Zhenglin Wang, Zekun Xi, Runnan Fang, Linhai Zhang, Yulan  
 624 He, Deyu Zhou, Pengjun Xie, and Fei Huang. Webwalker: Benchmarking llms in web traversal,  
 625b. URL <https://arxiv.org/abs/2501.07572>.

626

627 Qingyun Wu, Gagan Bansal, Jieyu Zhang, Yiran Wu, Beibin Li, Erkang Zhu, Li Jiang, Xiaoyun Zhang,  
 628 Shaokun Zhang, Jiale Liu, et al. Autogen: Enabling next-gen llm applications via multi-agent  
 629 conversation. *arXiv preprint arXiv:2308.08155*, 2023.

630

631 Tianle Xia, Liang Ding, Guojia Wan, Yibing Zhan, Bo Du, and Dacheng Tao. Improving complex  
 632 reasoning over knowledge graph with logic-aware curriculum tuning. In *Proceedings of the AAAI  
 633 Conference on Artificial Intelligence*, volume 39, pp. 12881–12889, 2025.

634

635 Huajian Xin, Daya Guo, Zhihong Shao, Zhizhou Ren, Qihao Zhu, Bo Liu, Chong Ruan, Wenda Li,  
 636 and Xiaodan Liang. Deepseek-prover: Advancing theorem proving in llms through large-scale  
 637 synthetic data. *arXiv preprint arXiv:2405.14333*, 2024.

638

639 An Yang, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chengyuan Li,  
 640 Dayiheng Liu, Fei Huang, Haoran Wei, et al. Qwen2.5 technical report. *arXiv preprint  
 641 arXiv:2412.15115*, 2024.

642

643 Zhilin Yang, Peng Qi, Saizheng Zhang, Yoshua Bengio, William W Cohen, Ruslan Salakhutdinov,  
 644 and Christopher D Manning. Hotpotqa: A dataset for diverse, explainable multi-hop question  
 645 answering. *arXiv preprint arXiv:1809.09600*, 2018.

646

647 Shunyu Yao, Jeffrey Zhao, Dian Yu, Nan Du, Izhak Shafran, Karthik Narasimhan, and Yuan Cao.  
 648 React: Synergizing reasoning and acting in language models. In *International Conference on  
 649 Learning Representations (ICLR)*, 2023.

650

651 Qinghao Ye, Haiyang Xu, Guohai Xu, Jiabo Ye, Ming Yan, Yiyang Zhou, Junyang Wang, Anwen Hu,  
 652 Pengcheng Shi, Yaya Shi, et al. mPLUG-Owl: Modularization empowers large language models  
 653 with multimodality. *CoRR*, abs/2304.14178, 2023.

648 Peilin Zhou, Bruce Leon, Xiang Ying, Can Zhang, Yifan Shao, Qichen Ye, Dading Chong, Zhiling  
649 Jin, Chenxuan Xie, Meng Cao, et al. Browsecmp-zh: Benchmarking web browsing ability of  
650 large language models in chinese. *arXiv preprint arXiv:2504.19314*, 2025.

651

652 He Zhu, Tianrui Qin, King Zhu, Heyuan Huang, Yeyi Guan, Jinxiang Xia, Yi Yao, Hanhao Li,  
653 Ningning Wang, Pai Liu, Tianhao Peng, Xin Gui, Xiaowan Li, Yuhui Liu, Yuchen Eleanor Jiang,  
654 Jun Wang, Changwang Zhang, Xiangru Tang, Ge Zhang, Jian Yang, Minghao Liu, Xitong Gao,  
655 Jiaheng Liu, and Wangchunshu Zhou. Oagents: An empirical study of building effective agents,  
656 2025. URL <https://arxiv.org/abs/2506.15741>.

657

658

659

660

661

662

663

664

665

666

667

668

669

670

671

672

673

674

675

676

677

678

679

680

681

682

683

684

685

686

687

688

689

690

691

692

693

694

695

696

697

698

699

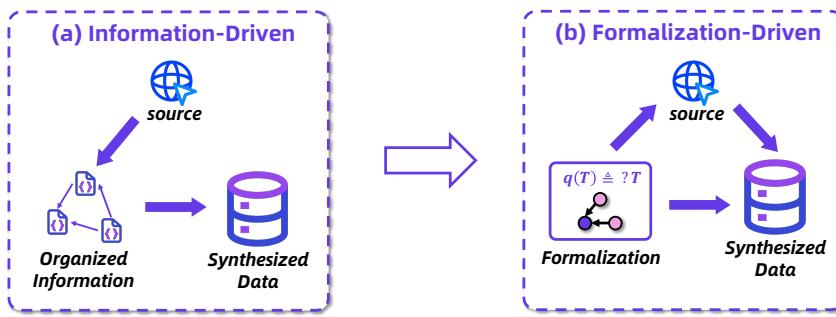
700

701

## 702 A DECLARATION ON THE USE OF LLMs

704 We affirm that the use of large language models in preparing this manuscript was strictly confined  
 705 to language-related assistance, including sentence refinement and grammatical correction. All  
 706 substantive content was independently authored by the authors and subsequently subjected to rigorous  
 707 review and verification following any LLM-assisted edits. In conducting the experiments, LLMs were  
 708 employed exclusively for legitimate academic research purposes, with no inappropriate applications.  
 709 Detailed experimental settings are provided in Sec. 4 of this paper. Beyond the aforementioned  
 710 language and experimental uses, no other reliance on LLMs was involved in this work.

## 712 B ILLUSTRATION OF FORMALIZATION-DRIVEN



724 Figure 6: Data synthesis paradigm shift from information-driven to formalization-driven. “Source”  
 725 stands for information sources such as the internet and databases. “Data” represents the synthesized  
 726 QA data. (a) Previous methods retrieve and organize collected information in advance, then synthesize  
 727 data according to the information structures. (b) Our method establishes the task formalization first,  
 728 then collects information, and synthesizes QA data based on the formalization.

730 Compared with the traditional *information-driven* paradigm (Figure 6(a), which first collects raw  
 731 information from various sources (e.g., the Internet and databases), organizes it into structured forms,  
 732 and subsequently synthesizes QA data according to these information structures, our *formalization-  
 733 driven* paradigm (Figure 6(b) begins by establishing a formalized representation of the target task  
 734 (e.g., logical or symbolic specification). Guided by this formalization, we then acquire relevant  
 735 information from sources and synthesize QA data directly in alignment with the established formal  
 736 specification. This shift in paradigm emphasizes precise task modeling prior to information retrieval,  
 737 enabling more controlled and consistent data generation.

738 Our data synthesis framework presents a foundational methodology for constructing training data for  
 739 intelligent agents, featuring two key innovations: **task formalization** and **agent-driven synthesis**.  
 740 By explicitly modeling tasks as structured, formal representations and leveraging proxy agents  
 741 to synthesize data, this work provides a systematic approach to address the critical challenge of  
 742 generating training data that transcends the complexity and unpredictability of naturally occurring  
 743 human-centric environments. Below, we discuss the broader implications for agent research.

744 **Implications in Agent Training Data Synthesis** Traditional approaches to training agents often  
 745 rely on datasets derived from human-generated interactions, which are inherently limited in diversity,  
 746 scalability, and controllability. We emphasize that effective agent training requires **explicit formal-  
 747 ization of task structures**—a prerequisite for achieving precise control over data properties. By  
 748 decoupling task definitions from data generation, the framework enables:

- 750 • *Targeted Complexity Management*: Tasks can be systematically parameterized to adjust  
 751 difficulty, modality, or compositional structure, ensuring agents are exposed to controlled  
 752 gradients of challenge. This contrasts with ad-hoc methods that risk overfitting to biases in  
 753 natural data or failing to stress-test edge cases.
- 754 • *Quality Assurance*: Formal task models act as a “specification” for data synthesis, reducing  
 755 noise and ensuring consistency. This is critical for applications where reliability and safety  
 756 are paramount, such as autonomous systems or medical AI.

756

- *Scalable Data Generation*: Agent-driven synthesis eliminates the need for laborious manual  
757 annotation or heuristic-based pipelines by directly translating formal task representations  
758 into training instances. This reduces computational overhead while preserving fidelity to the  
759 task’s intended design.

760

761 **Implications for AI Research and Development** Our architecture provides insights for advancing  
762 AI systems:

763

764

- *Beyond Human-Level Complexity*: By formalizing tasks independent of human behavioral  
765 priors, the framework enables training data to exceed the implicit constraints of natural  
766 data. This opens pathways to train agents for domains requiring superhuman reasoning (e.g.,  
767 advanced scientific modeling, combinatorial optimization).
- *Cross-Domain/Task Generalization*: Formal task representations abstract away domain-  
768 specific noise, allowing agents to learn invariant principles applicable across diverse contexts.

769

770

## C PROOF OF PROPOSITION 1

771

772 Here we provide formal proof of Proposition 1.

773

774

775 *Proof.* Let  $x$  be an element of  $R(S_1) \cup R(S_2)$ . By Equation 1, there exists either a  $y_1 \in S_1$  such  
776 that  $(y_1, x) \in R$  or  $(x, y_1) \in R$ , or a  $y_2 \in S_2$  such that  $(y_2, x) \in R$  or  $(x, y_2) \in R$ . Consequently,  
777 there exists a  $y \in S_1 \cup S_2$ , e.g.,  $y_1$  or  $y_2$ , such that  $(y, x) \in R$  or  $(x, y) \in R$ . Thus, we have  
778  $x \in R(S_1 \cup S_2)$ , and hence  $R(S_1) \cup R(S_2) \subseteq R(S_1 \cup S_2)$ .

779 Conversely, let  $z$  be an element of  $R(S_1 \cup S_2)$ . Then there exists a  $y \in S_1 \cup S_2$  such that  $(y, z) \in R$  or  
780  $(z, y) \in R$ . If  $y \in S_1$ , then  $z \in R(S_1)$ ; if  $y \in S_2$ , then  $z \in R(S_2)$ . In either case,  $z \in R(S_1) \cup R(S_2)$ .  
781 Therefore,  $R(S_1 \cup S_2) \subseteq R(S_1) \cup R(S_2)$ .

782 Combining both directions, we conclude that:

783

$$R(S_1) \cup R(S_2) = R(S_1 \cup S_2).$$

784 Thus, we end proof of the Proposition. □

785

786

## D DATA STATISTICS

787

788 We analyze the domain distributions of our dataset.  
789 The domain distribution of our dataset demonstrates  
790 rather comprehensive coverage across multiple thematic  
791 areas, as visualized in Figure 7. Our construction  
792 of seed tasks leads to questions about various  
793 topics and entities. Our agentic expansion further  
794 strengthens these benefits. The dataset achieves sig-  
795 nificant diversity through its balanced representation  
796 of major domains such as Sports, Politics, and  
797 Entertainment.

800 This deliberate design ensures our dataset not only  
801 avoids over-reliance on any single domain but also  
802 maintains sufficient sample density across diverse  
803 topics. The empirical balance between breadth and  
804 depth enables robust training of a domain-agnostic  
805 information-seeking agent. Such characteristics po-  
806 sition our dataset as particularly suitable for train-  
807 ing multi-domain IS tasks and fostering interdisciplinary  
808 research.

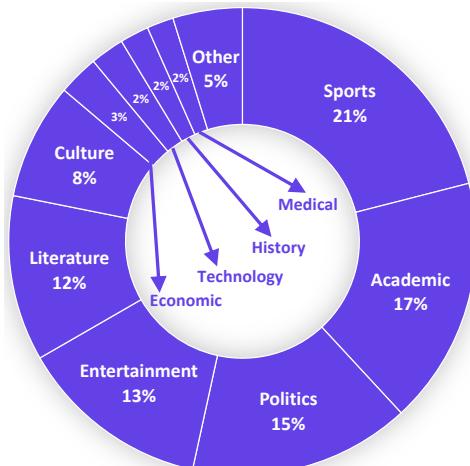


Figure 7: Domain distribution.

810  
811 E COMPARED DATASETS  
812813 We compare our synthesized dataset with several datasets:  
814815 

- 816 • WebWalkerQA employs random walks over interlinked URLs to synthesize questions based  
817 on the visited webpages (Wu et al., 2025b). The dataset includes both single-source questions,  
818 generated from a single visited URL, and multi-source questions, which are constructed  
819 using information aggregated from multiple visited URLs.
- 820 • E2HQA is a dataset introduced by WebDancer (Wu et al., 2025a), where simple questions are  
821 systematically rewritten into more complex, challenging ones.
- 822 • MHQA is a composite dataset that integrates existing single-hop and multi-hop question-  
823 answering datasets. The majority of the questions are annotated by humans.

824 F AGENT TRAINING  
825826 To train our information-seeking agent, similar to WebDancer (Wu et al., 2025a), we implement  
827 supervised fine-tuning (SFT) followed by reinforcement learning (RL).  
828829 In SFT, given a trajectory in a sequence of tokens  $\mathcal{T} = (\tau_1, \alpha_1, o_1, \dots, \tau_n, \alpha_n, o_n)$ , we mask out loss  
830 from observation leading to loss:  
831

832 
$$L = -\frac{1}{\sum_{i=1}^{|\mathcal{T}|} \mathbb{I}[x_i \in o]} \sum_{i=1}^{|\mathcal{T}|} \mathbb{I}[x_i \in o] \cdot \log \pi_\theta(x_i \mid x_{<i}) \quad (8)$$

833 where  $\pi_\theta$  is the model to be trained. Later in RL, we further optimize  $\pi_\theta$  based on the GRPO  
834 algorithm (Shao et al., 2024). For a question-answer pair  $(q, a)$ , we sample rollouts  $\{y_i\}_i^{|\mathcal{G}|}$  and  
835 update the policy model by:

836 
$$\begin{aligned} \mathcal{J}(\theta) = & \mathbb{E}_{q \sim \mathcal{D}, \{y_i\}_{i=1}^{|\mathcal{G}|} \sim \pi_{\theta_{\text{old}}}(\cdot \mid \text{context})} \\ & \left[ \frac{1}{\sum_{i=1}^{|\mathcal{G}|} |y_i|} \sum_{i=1}^{|\mathcal{G}|} \sum_{t=1}^{|y_i|} \min \left( r_{i,t}(\theta) \hat{A}_{i,t}, \text{clip} \left( r_{i,t}(\theta), 1 - \varepsilon_{\text{low}}, 1 + \varepsilon_{\text{high}} \right) \hat{A}_{i,t} \right) \right] \quad (9) \\ & r_{i,j}(\theta) = \frac{\pi_\theta(o_i \mid q_i, o_{i,<t})}{\pi_{\theta_{\text{old}}}(o_i \mid q_i, o_{i,<t})}, \quad \hat{A}_{i,j} = \frac{R_i - \text{mean}(\{R_i\})}{\text{std}(\{R_i\})}, \end{aligned}$$

837 where *context* includes all the model completions and tool responses.  $\varepsilon$  is the clipping range of the  
838 importance sampling ratio  $r_{i,t}(\theta)$ .  $\hat{A}_{i,t}$  is an estimator of the advantage of the  $i$ -th rollout at  $t$ -th step.  
839840 G AGENT DETAILS  
841842 Following Wu et al. (2025a), WebShaper uses two tools, *search* and *visit*, which are regarded as  
843 fundamental to the information seeking process (Zhu et al., 2025):  
844845 

- 846 • **Search** interfaces with the Google search engine to retrieve relevant documents given natural  
847 language queries. It supports multiple queries in parallel and returns the top-10 results for  
848 each query, where each result includes a title, a snippet, and the corresponding URL.
- 849 • **Visit** enables targeted extraction from specific web pages. Each page is paired with a  
850 designated visit goal. The full content of the page is first retrieved using Jina (Jina.ai,  
851 2025), after which a summarization model (Qwen-2.5-72B in our implementation) extracts  
852 information relevant to the specified goal.

853 H TRAINING DETAILS  
854855 H.1 SFT  
856857 For SFT, we use a batch size of 32 and a learning rate of 5e-6, warmup plus cosine decay schedule.  
858 We also apply a weight decay of 0.1.  
859

864 H.2 RL

865

866 For RL training (Sheng et al., 2025), each group consists of 8 rollouts. The temperature is 1.0,  
 867  $top_p = 1.0$ , the batch size is 128, the mini batch size is 32, and the learning rate is 1e-6.

868

869 H.2.1 CASE STUDY

870

### 871 Question In Natural Language

872 **Question:** What is the title of the section, where the section is written by an author who also authored a  
 873 scholarly article analyzing contact between Medieval Norse and Native North Americans published in a peer-  
 874 reviewed archaeology journal, which additionally published another article that analyzes Lake Mohave artifacts  
 875 and Pleistocene lake levels?

876 **Answer:** Thule Prehistory of Canada.

877

878

### 879 Question In Formalization

```
[ [V@X, hasTitle, V@T], [V@X, writtenBy, V@Y],  

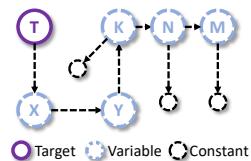
  [V@Y, hasAuthor, V@K], [V@K, publishIn, V@N], [V@N, publish, V@M],  

  [V@K, analyzeContactBetween, C@Medieval Norse and Native North Americans],  

  [V@N, isA, C@A peer-reviewed archaeology journal],  

  [V@M, analyze, C@Lake Mohave artifacts and Pleistocene lake level] ]
```

### 880 Question In Graph



881

882

883

884

885 Figure 8: Case studies of our synthesized data. We show a question in natural language, our  
 886 formalization, and a graph respectively.

887

888 We present a representative case study in Figure 8. Compared with linear structure and sequential  
 889 structure, our synthesized data has no problems of redundancy and reasoning shortcuts. The model  
 890 should strictly seek information and reason alongside all the variables to find the answer. There are  
 891 no constants directly connected to the target variable  $T$  or variables close to it. Besides, there are no  
 892 constants connected to other constants.

893

894 Moreover,  $R$ -Union effects well in our data. The underlined FP is a summarization of distributed  
 895 web contents, leading to more difficulty in resolving the variables  $K$ ,  $N$ , and  $M$ . Benefiting from  
 896 the formalization, our data contains a variety of IS forms, which can fully stimulate the different IS  
 897 capabilities of the model.

898

899

900

901

902

903

### 904 Question In Natural Language

905 **Question:** "Strange Stories from a Chinese Studio" is a collection of classical Chinese short stories written by the  
 906 Qing Dynasty novelist Pu Songling. The earliest manuscript copies were already in circulation during the Kangxi  
 907 reign of the Qing Dynasty, and the collection comprises over four hundred short stories in total. In Volume Ten of  
 908 "Strange Stories from a Chinese Studio," there is a story titled "The Green-Clothed Girl." In this story, how many  
 909 sentences did the scholar Yu Jing speak with her?

910

911

### 912 Question In Formalization

```
[ [C@SSCS, isA, C@Classic story], [C@SSCS, writtenBy, C@Pu Songling],  

  [C@SSCS, inCirculation, C@Qing Dynasty],  

  [C@SSCS, comprises, C@over 400 stories],  

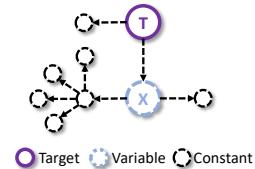
  [V@X, isInTenVolume, C@SSCS],  

  [V@X, isTitled, C@The Green-Clothed Girls],  

  [V@X, hasSentences, V@T],  

  [V@T, happenedBetween, C@Yu Jing_The Green-Clothed Girl] ]
```

### 913 Question In Graph



914

915

916

917

918 Figure 9: Case comparison. "SSCS" stands for "Strange Stories from a Chinese Studio".

919 We compare a representative example shown by KIMI-Researcher (Kimi, 2025), illustrated in Figure 9.  
 920 The case includes redundant information, such as multiple constants connected to "SSCS", which  
 921 contribute little to answering the question. Additionally, a reasoning shortcut is observed that directly  
 922 connects to the target variable. Despite the apparent complexity, the underlying reasoning structure is  
 923 relatively simple, consisting of a single-hop reasoning path.

918  
 919 Table 3: **SFT Data Comparison** on GAIA benchmarks. The best results among all backbones are in  
 920 **bolded**.  
 921  
 922  
 923

Backbone	Dataset	GAIA			
		Level 1	Level 2	Level 3	Avg.
Qwen-2.5-32B	WebWalkerQA	43.5	30.7	0.0	32.0
	E2HQA	56.4	36.5	0.0	39.8
	MHQA	43.5	36.5	8.3	35.9
	<b>WebShaper</b>	56.4	40.3	16.6	<b>43.6</b>
Qwen-2.5-72B	WebWalkerQA	53.8	36.5	0.0	38.8
	E2HQA	61.5	38.4	16.6	44.6
	MHQA	56.4	44.2	0.0	43.6
	<b>WebShaper</b>	56.4	48.0	0.0	<b>45.6</b>
QwQ-32B	WebWalkerQA	66.6	38.4	8.3	45.6
	E2HQA	58.9	42.3	16.6	45.6
	MHQA	51.2	44.2	0.0	41.7
	<b>WebShaper</b>	69.2	50.0	16.6	<b>53.3</b>

935  
 936 **I DETAILED DATA COMPARISON RESULTS**  
 937

938 As shown in Table 3, our proposed WebShaper method consistently achieves the highest average  
 939 performance across different backbones on the GAIA benchmarks. In particular, WebShaper out-  
 940 performs other datasets in most settings, with the best results highlighted in bold. For example,  
 941 with the Qwen-2.5-32B backbone, WebShaper achieves an average score of 43.6, surpassing com-  
 942 peting datasets by a significant margin. Similarly, for Qwen-2.5-72B and QwQ-32B backbones,  
 943 WebShaper reaches 45.6 and 53.3 respectively, demonstrating strong generalization capabilities  
 944 across model sizes and difficulty levels (Level 1, Level 2, Level 3). These results clearly highlight the  
 945 robustness and superiority of our approach in handling diverse and challenging evaluation settings.  
 946  
 947  
 948  
 949  
 950  
 951  
 952  
 953  
 954  
 955  
 956  
 957  
 958  
 959  
 960  
 961  
 962  
 963  
 964  
 965  
 966  
 967  
 968  
 969  
 970  
 971