Deep&WideSearch: Benchmarking Depth and Breadth in Real-World Info-Seeking

Tian Lan, Bin Zhu, Qianghuai Jia, Junyang Ren, Haijun Li, Longyue Wang*, **Zhao Xu, Weihua Luo, Kaifu Zhang**
Alibaba International Digital Commerce
* Corresponding Author: Longyue Wang

Abstract

Current search agents fundamentally lack the ability to simultaneously perform *deep* reasoning over multi-hop retrieval and *wide*-scale information collection—a critical deficiency for real-world applications like comprehensive market analysis and business development. To bridge this gap, we introduce DeepWideSearch, the first benchmark explicitly designed to evaluate agents to integrate depth and width in information seeking. In DeepWideSearch, agents must process a large volume of data, each requiring deep reasoning over multi-hop retrieval paths. Specifically, we propose two methods to converse established datasets, resulting in a curated collection of 220 questions spanning 15 diverse domains. Extensive experiments demonstrate that even state-of-the-art agents achieve only 2.39% average success rate on DeepWideSearch, highlighting the substantial challenge of integrating depth and width search in information-seeking tasks. Furthermore, our error analysis reveals four failure modes: lack of reflection, overreliance on internal knowledge, insufficient retrieval, and context overflow—exposing key limitations in current agent architectures. We publicly release DeepWideSearch to catalyze future research on more capable and robust information-seeking agents.

https://github.com/AIDC-AI/Marco-DeepWideSearch-Agent

https://huggingface.co/datasets/AIDC-AI/DeepWideSearch

1. Introduction

Large Language Models (LLMs) with advanced reasoning capabilities [Achiam et al., 2023, Liu et al., 2024, Guo et al., 2025] have driven substantial progress across a wide range of natural language tasks. Building on these advances, LLM-based agents that equipped with planning, tool use, and multi-step reasoning capabilities [Xi et al., 2025, Gao et al., 2025]—have achieved strong performance on complex real-world challenges, including computer operation [Wang et al., 2025], deep research [Du et al., 2025], and information seeking [Mialon et al., 2023, Wei et al., 2025].

So far, existing benchmarks for evaluating agents can be systematically categorized along two critical dimensions (Figure 1): search width (measured by the number of information units to be searched) and search depth (measured by average search steps for each unit), revealing four distinct categories: (1) *Low width, high depth* benchmarks (e.g., GAIA [Mialon et al., 2023],

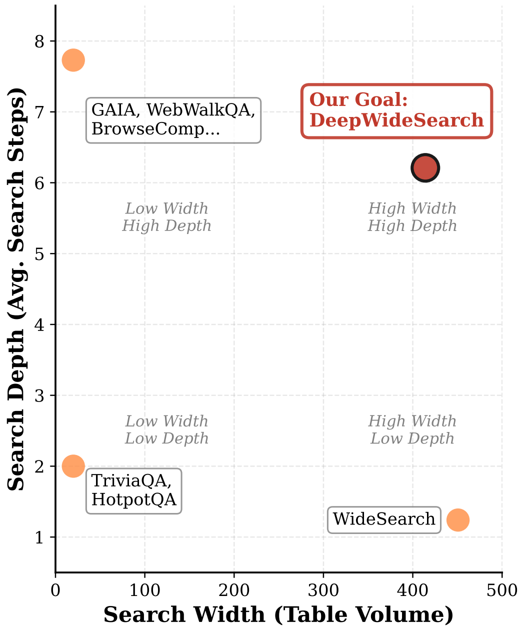
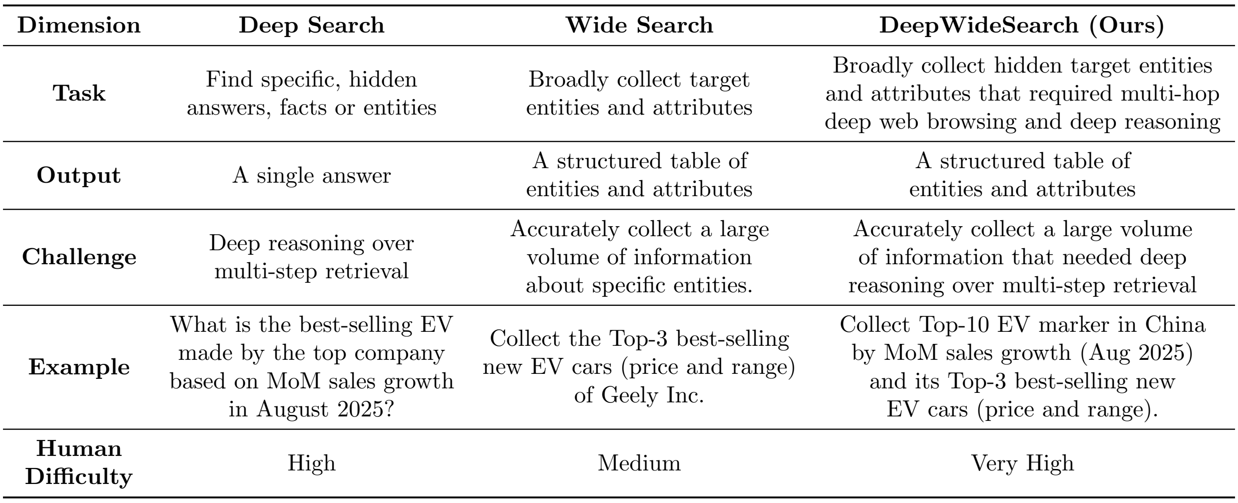


Figure 1 | The comparison of existing benchmarks on search width and depth.

© 2025 Alibaba. All rights reserved



Dimension	Deep Search	Wide Search	DeepWideSearch (Ours)
Task	Find specific, hidden answers, facts or entities	Broadly collect target entities and attributes	Broadly collect hidden target entities and attributes that required multi-hop deep web browsing and deep reasoning
Output	A single answer	A structured table of entities and attributes	A structured table of entities and attributes
Challenge	Deep reasoning over multi-step retrieval	Accurately collect a large volume of information about specific entities.	Accurately collect a large volume of information that needed deep reasoning over multi-step retrieval
Example	What is the best-selling EV made by the top company based on MoM sales growth in August 2025?	Collect the Top-3 best-selling new EV cars (price and range) of Geely Inc.	Collect Top-10 EV marker in China by MoM sales growth (Aug 2025) and its Top-3 best-selling new EV cars (price and range).
Human Difficulty	High	Medium	Very High

Figure 2 | The detailed comparison among deep search, wide search benchmarks and our proposed DeepWideSearch.

BrowseComp [Wei et al., 2025]), which focus on intricate deep reasoning over multi-hop retrieval for searching target answers; (2) *Low width, low depth* benchmarks (e.g., TriviaQA, HotpotQA), which address simple fact-finding tasks; (3) *High width, low depth* benchmarks (e.g., WideSearch [Wong et al., 2025] and PaSa [He et al., 2025]), which emphasize broad information collection about specific questions; and critically, (4) *High width, high depth* tasks, which collect extensive information that required deep reasoning—a critical capability for real-world applications like comprehensive market analysis and business development but entirely unaddressed by current benchmarks. For instance, as shown in Figure 2, the case *"identifying the Top-10 EV maker in China by MoM sales growth (Aug 2025) and its Top-3 best-selling new EV cars (price and range)"* exemplifies this challenge. It requires agent to gather a large volume of candidates, *i.e.*, *EV makers*, to fill the result table through wide-scale search, and verify each candidate by performing deep reasoning, a combinatorial complexity that exceeds both the scope of width-focused evaluations and the scale of depth-focused benchmarks.

To address this critical evaluation gap, we introduce DeepWideSearch, the first benchmark explicitly designed to evaluate the capability of agents in deep and wide information seeking. Since it is challenging to construct deep and wide search instances even with human annotation, we develop two methods for conversing established datasets: (1) *Deep2Wide Conversion*, which extends deep search benchmarks (e.g., GAIA and BrowseComp) by augmenting their information scope through human-annotated table schemas; and (2) *Wide2Deep Conversion*, which enhances wide search queries by replacing explicit entities with synthesized complex sub-questions that necessitate multi-hop search steps. Both approaches integrate rigorous human validation protocols to ensure data quality while maintaining the combinatorial complexity inherent in real-world information-seeking scenarios. The final benchmark comprises 220 meticulously curated questions spanning 15 diverse domains, featuring both Chinese and English queries with human-verified ground truths, with 85 instances derived from Deep2Wide and 135 from Wide2Deep construction methods.

We conduct comprehensive experiments across state-of-the-art LLMs and agent systems on DeepWideSearch. Our results demonstrate that even the most advanced agent systems achieve only 2.39% average success rate on DeepWideSearch, highlighting the substantial difficulty of this kind of information seeking task. Notably, while agent frameworks consistently improve core entity identification (e.g., +15.91 absolute percentage points in Core Entity Accuracy), they exhibit limited efficacy in wide-scale information collection, frequently underperforming their LLMs counterparts using internal knowledge. Through systematic error analysis, we identify four fundamental failure modes: (1) lack of effective reflection mechanisms when encountering problematic search trajectories; (2) overreliance on parametric internal knowledge leading to outdated or inaccurate information; (3) insufficient retrieval despite accessing relevant webpages; and (4) context overflow exceeding current agent architectue limitations. These empirical findings expose key limitations of current agent architecture for the deep and wide information-seeking task. To facilitate further research in this critical domain, we have publicly released the DeepWideSearch benchmark, including datasets and evaluation codebase.

2. Related Work

2.1. LLM-based Search Agents

The emergence of LLM-based agent systems has enabled sophisticated information-seeking capabilities, with frameworks ranging from closed-source implementations (e.g., OpenAI Deep Research) to open-source platforms (e.g., WebAgent [Wu et al., 2025b] and Cognitive Kernel-Pro [Fang et al., 2025b]). These systems have demonstrated proficiency in numerous application domains, including computer-use agents, deep research for complex problem investigation [Han et al., 2025], and multi-step information retrieval through tool use [Xi et al., 2025]. Among these applications, information-seeking agents represent a critical frontier impact real-world utility. Current research in this domain primarily addresses five technical challenges: (1) agentic system architecture design [Zhang et al., 2025a, Zhou et al., 2025a, Xia et al., 2025, Fang et al., 2025a], (2) synthetic data generation for complex scenarios [Wu et al., 2025a, Li et al., 2025, Tao et al., 2025], (3) optimization techniques for retrieval efficiency [Zhang et al., 2025b, Fan et al., 2025, Sun et al., 2025], (4) knowledge management for multi-hop reasoning [Zhang et al., 2025a, Xu et al., 2025], and (5) evaluation methodologies for performance assessment [Zhuge et al., 2025, Gou et al., 2025].

2.2. Benchmarks for LLM-based Agents

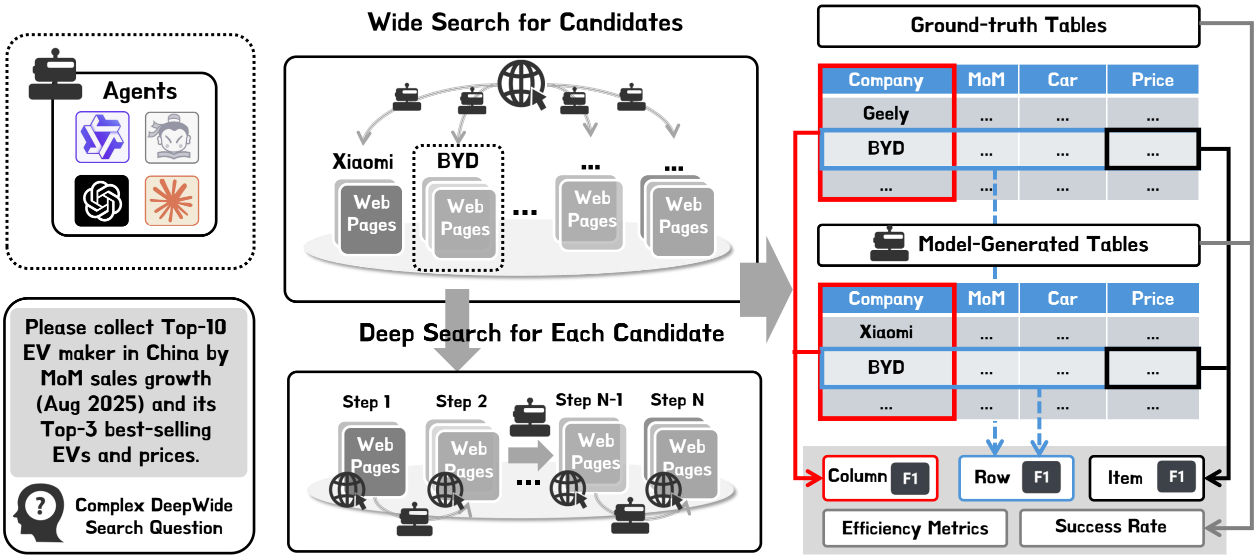
Existing evaluation frameworks for information-seeking agents primarily target two distinct capabilities: (1) *Depth* in multi-hop reasoning, measured by benchmarks like GAIA [Mialon et al., 2023] and BrowseComp [Wei et al., 2025] for general complex reasoning, and domain-specific variants in healthcare [Chen et al., 2025] and E-commerce [Lyu et al., 2025]; (2) *Width* in information collection, assessed by WideSearch [Wong et al., 2025] for comprehensive retrieval of atomic information, and PaSa [He et al., 2025] and SPAR [Shi et al., 2025] for academic literature retrieval. Crucially, no existing benchmark captures the *combinatorial complexity* inherent in real-world information-seeking tasks that simultaneously demand extensive exploration (width) and intricate multi-step reasoning (depth). This fundamental gap in evaluation methodology has prevented meaningful progress toward agents capable of handling the complex real-world information-seeking. To address this limitation, we propose DeepWideSearch, the first benchmark explicitly designed to evaluate the capability of agents in the deep and wide information-seeking task.

3. Task Formulation

As shown in Figure 3, DeepWideSearch establishes an evaluation framework that explicitly captures the *combinatorial complexity* of real-world information-seeking tasks—requiring agents to perform *deep* reasoning and *wide*-scale information collection. The evaluation metrics (Column F1, Row F1, Item F1, and Success Rate) illustrated in Figure 3 will be formally described in Section 4.4.

Input Formally, each task in DeepWideSearch is defined as a tuple (Q, C): (1) **Question Q** represents a complex natural language query for deep and wide information seeking; and (2) **Columns $C = \{c_i\}_{i=1}^N$** define the table schema as a set of N attributes and constraints need to be collected and verified, such as EV price and MoM scales growth in Figure 3 (right).

Output As shown in Figure 3 (medium), agents are required to generate a structured tabular response R by performing wide search for gathering numerous candidates and deep search for the verification of each candidate.


Agents
Please collect Top-10 EV maker in China by MoM sales growth (Aug 2025) and its Top-3 best-selling EVs and prices.
Complex DeepWide Search Question
Wide Search for Candidates
Xiaomi
BYD
...
...
Web Pages
Deep Search for Each Candidate
Step 1
Step 2
Step N-1
Step N
Web Pages
Ground-truth Tables
Model-Generated Tables
Column F1
Row F1
Item F1
Efficiency Metrics
Success Rate

Company	MoM	Car	Price
Geely	...	...	...
BYD	...	...	...
...	...	...	...

Company	MoM	Car	Price
Xiaomi	...	...	...
BYD	...	...	...
...	...	...	...

Figure 3 | Task formulation of DeepWideSearch task. The evaluation metrics (highlighted in red) are detailed in Section 4.4.

4. Methodology of Dataset Construction

Constructing DeepWideSearch instances from scratch presents significant challenges due to the substantial human effort. To address this challenge while maintaining methodological rigor, we propose two methods to converse established datasets into deep and wide search questions: (1) Deep2Wide Conversion and (2) Wide2Deep Conversion. Both methodologies are complemented by human annotation procedures to ensure the quality.

4.1. Convert Deep Search Datasets (Deep2Wide)

Existing deep search benchmarks such as GAIA [Mialon et al., 2023], BrowseComp [Wei et al., 2025] and BrowseComp-zh [Zhou et al., 2025b] require agents to employ multi-hop web browsing and deep reasoning to identify target answers. Building upon these resources, we develop the Deep2Wide conversion methodology by expanding the scope of searched information. As illustrated in Figure 4 (Top), our approach follows a three-stage pipeline inspired by WideSearch [Wong et al., 2025]: (1) **Core Entity Filtering**: We sample 80 Chinese questions from BrowseComp-zh and 20 English questions from BrowseComp, filtering out instances where answers are unsuitable as core entities (*e.g.*, dates and numerical values). For example, as shown in Figure 5, *Dan Lin* is the core entity of the deep search question; (2) **Table Schema Definition**: Human annotators design structured table schemas by defining relevant information about the core entities; (3) **Comprehensive Annotation**: Annotators

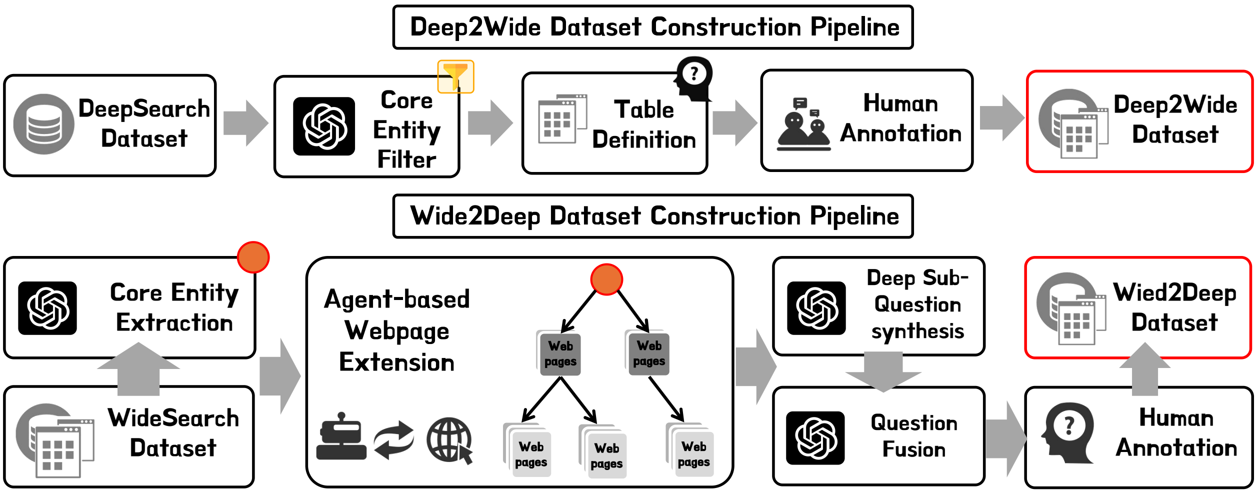


Figure 4 | The pipelines of our proposed Deep2Wide and Wide2Deep data construction methods.

perform exhaustive web searches to populate the tables. Each instance requires approximately 30 minutes of human annotation time, ensuring high-quality and verified data. Following a design similar to that of the WideSearch benchmark [Wong et al., 2025], we incorporated timestamps into each question to ensure that the answers remain invariant over time.

> **Deep Search Question**: There is a Chinese athlete who has achieved remarkable success in a ball sport. He was the first player in his discipline to successfully defend his title at a major competition and has won multiple world championship titles. His sport underwent rule changes in the early 21st century, and he became the first men's singles Olympic champion under the new rules.
> **Core Entity**: Dan Lin.
> **Table Schema**: Competition records of Dan Lin from 2010 to 2020 including the following columns: Date, Tournament Name, Level, Event, Result, and Match Details (including opponent, score, and win/loss outcome).

Figure 5 | One deep search question in BrowseComp-ZH.

4.2. Convert Wide Search Datasets (Wide2Deep)

Given that WideSearch [Wong et al., 2025] represents the publicly available dataset providing human-annotated tabular answers for wide-scale information-seeking, we develop the Wide2Deep conversion methodology to transform these wide search queries by introducing complexity in entity identification. This approach reuse the valuable human-annotated table in WideSearch while enhancing the deep reasoning requirements. Inspired by WebWalker [Wu et al., 2025b], we implement a human-in-the-loop pipeline (Figure 3, bottom) comprising five stages: (1) **Entity Extraction**: Advanced LLMs identify core entities in 160 English and Chinese WideSearch questions, similar to the core entity in the deep search benchmark (Figure 5); (2) **Deep Sub-Question Synthesis**: Following prior work [Li et al., 2025, Tao et al., 2025], a web search agent are implemented to recursively traverse official websites about core entities and collecting their rich entity information. Then, a complex sub-question is generated based on these rich information, adhering to two critical constraints: (a) Uniqueness: The answer to the question must be a single, well-defined entity; (b) Complexity: Direct derivation of the entity from the question must require at least one additional web search step; (3) **Question Fusion**: Claude-sonnet-4 fuses the deep sub-question with the original wide search query; and (4) **Human Annotation**: A team of seven master's-level annotators validates and refines the synthesized questions to ensure uniqueness, complexity, and linguistic naturalness. This process requires approximately 40 minutes of human annotation per instance, maintaining the high-quality standards essential for a rigorous benchmark. The prompts of core entity extraction, deep sub-question synthesis and question fusion are placed at Appendix C.

4.3. Data Statistics

Table 1 provides a comprehensive comparison of our DeepWideSearch benchmark against existing datasets across multiple dimensions. Our benchmark demonstrates significantly higher search complexity compared to prior work, with an average table volume of 414.10 information units, substantially exceeding deep search benchmarks like GAIA and BrowseComp. Crucially, DeepWideSearch requires 4.21 average search steps to identify core entities—nearly 4× more complex than WideSearch (1.24). The dataset spans 15 diverse domains, covering both English and Chinese queries, with 220 carefully curated instances (85 from Deep2Wide, 135 from Wide2Deep). These statistics empirically validate the deep and wide attributes of our proposed DeepWideSearch. Cases and more details about the data in Table 1 can be found in Appendix A.

Benchmarks	Domains	Data Size	Avg. Sample Per Domain	Table Volume	Avg. Steps Search Entity	Lang.
TriviaQA [Joshi et al., 2017]	-	95K	-	1	≈ 1	EN
HotpotQA [Yang et al., 2018]	-	113K	-	1	≈ 2	EN
GAIA [Mialon et al., 2023]	-	103	-	1	7.73	EN
BrowseComp [Wei et al., 2025]	9	1266	126.6	1	-	EN
BrowseComp-zh [Zhou et al., 2025b]	11	289	26.27	1	-	ZH
WideSearch [Wong et al., 2025]	14	200	12.80	450.67	1.24	EN,ZH
Our Proposed DeepWideSearch						
Deep2Wide	15	85	7.08	247.74	3.22	EN,ZH
Wide2Deep	13	135	10.38	518.84	4.55	EN,ZH
Overall	15	220	14.67	414.10	4.21	EN,ZH

Table 1 | Data statistics comparison across benchmarks. GAIA refers to the text-only split.

4.4. Evaluation Metrics of DeepWideSearch

As shown in Figure 3, we evaluate agent performance on DeepWideSearch along three complementary axes: Depth, Width, and Efficiency.

Depth Evaluation The depth dimension evaluate the capability of agents to correctly identify target entities through deep reasoning over multi-hop retrieval. Following previous works [Wei et al., 2025, Mialon et al., 2023], we introduce the **Column-F1** metric. As shown in Figure 3, Column-F1 is computed as the F1 score over the unique columns in the table. These unique columns correspond to the core attributes of entities (*i.e.*, rows) that uniquely identify them. Therefore, Column-F1 can be seen as the extension of the accuracy used in established deep search benchmarks, computing the precision of a group of entities (rows in the table). Higher Column-F1 scores indicate more precise entities identification across the entire table structure. Moreover, since our proposed two methods include the core entity of questions, we also introduce the **Core Entity Accuracy (CE Acc.)**, serving as an additional indicator of deep reasoning capability.

Width Evaluation The width dimension measures how comprehensively and accurately the agent retrieves all associated information units for entities (rows in the table). Building upon the evaluation framework of WideSearch [Wong et al., 2025], we assess performance at three granularities: (1) **Success Rate**: A binary metric indicating whether the agent's output table exactly matches the human-annotated ground truth (all rows, columns, and values identical); (2) **Row-level F1**: Computes precision, recall, and F1 scores at the row level (i.e., for each entity and its associated attributes), capturing whether the agent retrieves complete contextual information per entity; (3) **Item-level F1**: The finest-grained metric evaluating accuracy at the individual cell level, reflecting fidelity in retrieving atomic information units within the structured table.

Efficiency Evaluation To address the substantial computational costs inherent in web-scale tool usage (including search, browsing APIs), we further evaluate system efficiency through two metrics: (1) **Input/Output Token**: The total tokens consumed during reasoning and tool calls; (2) **Cost**: Estimated cost expenditure based on standard model inference API pricing during query resolution. These efficiency metrics are critical for real-world deployment considerations, particularly given the demanding requirements for extensive multi-round search and browsing.

To account for stochasticity in LLM-based agent behavior, we conduct four independent runs per question for each baseline system. For both depth and width metrics, we report three complementary statistics: (1) **Avg@4**: The mean performance across all four runs; (2) **Max@4**: The best performance observed across the four runs; and (3) **Pass@4**: The proportion of questions solved successfully in at least one run (only for Success Rate). This comprehensive evaluation protocol ensures robustness against sampling variance while also highlighting the system's peak performance potential.

5. Experiments

5.1. Experimental Setup

We evaluate three kinds of baselines on our proposed DeepWideBenchmark: (1) *Closed-source LLMs (without tool calls)*: OpenAI o3-mini, GPT-4o, GPT-5, Claude-sonnet 4, Gemini 2.5 Pro and Qwen-Max; (2) *Open-source LLMs (without tool calls)*: DeepSeek-V3/R1 [Guo et al., 2025, Liu et al., 2024], KIMI-K2 [Team et al., 2025], Qwen3 series [Yang et al., 2025]; and (3) *Open-source Agent Systems*: WebSailor [Li et al., 2025], Smolagents [Roucher et al., 2025] and OWL [Hu et al., 2025] are equipped with advanced GPT-5, Claude-sonnet-4 and Gemini-2.5-Pro backbone models. All agent systems utilized identical tools: (1) Google Search API; and (2) Webpage Visit tool. Since webpages in HTML format are often very lengthy, we use the same LLM in the agents to summarize the HTML into a concise summarization. The cost of this summarization process is also counted into the efficiency metrics. We utilized the official API endpoints of these LLMs with their default decoding parameter settings.

5.2. Main Results

The complete results are presented in Table 2. It can be found that most baselines demonstrate near-zero success rates, with only WebSailor (Gemini 2.5 Pro) and WebSailor (Claude Sonnet 4) exceeding 1-2% in Success Rate (Avg@4), confirming the inherent complexity of simultaneously handling deep reasoning and wide-scale information collection. Notably, Gemini 2.5 Pro emerges as the top-performing LLM, achieving the highest Column F1 (45.27%, Avg@4), Core Entity Accuracy (73.98%, Avg@4), and Pass@4 Success Rate (1.82%), even outperforming several agent systems. This exceptional performance indicates that Gemini 2.5 Pro possesses advanced reasoning capabilities

Model / System	**Success Rate (%)**		**Row F1 Score (%)**		**Item F1 Score (%)**		**Column F1 (%)**		**CE Acc. (%)**	
	Avg@4	Pass@4	Avg@4	Max@4	Avg@4	Max@4	Avg@4	Max@4	Avg@4	Pass@4
Closed-source LLMs										
OpenAI o3-mini	0.0	0.0	3.35	4.55	13.59	16.85	27.36	35.68	61.59	69.55
GPT-5	0.30	1.36	9.61	13.42	21.67	28.21	31.71	41.05	58.41	72.72
Claude Sonnet 4	0.9	0.9	7.31	8.97	19.94	23.38	32.63	40.16	57.95	64.09
Gemini 2.5 Pro	0.9	1.82	15.42	18.96	32.06	37.10	**45.27**	52.86	**73.98**	81.82
Qwen-Max	0.0	0.0	4.16	6.18	14.32	18.48	28.81	36.19	56.02	63.64
GPT-4o	0.0	0.0	4.18	7.01	11.86	16.41	19.66	27.07	54.20	63.64
Open-source LLMs										
DeepSeek-V3	0.23	0.45	6.52	9.99	19.08	24.32	31.26	39.56	60.68	69.09
DeepSeek-R1	0.28	0.45	10.72	14.39	25.01	30.56	38.42	47.77	66.93	75.45
KIMI-K2	0.34	0.91	7.74	11.92	20.44	27.54	31.48	41.83	64.32	73.18
Qwen3-235B-A22B	0.0	0.0	2.94	5.74	12.38	19.53	22.03	34.99	52.39	67.73
Qwen3-235B-A22B-Instruct	0.0	0.0	3.50	5.34	13.28	17.85	24.64	33.03	56.82	64.09
Qwen3-32B	0.0	0.0	2.28	3.67	12.05	16.26	26.37	35.97	54.66	66.36
Open-source Agent Framework with Advanced LLMs										
OWL (Gemini 2.5 Pro)	0.0	0.0	11.11	16.93	28.75	41.70	34.84	50.39	66.14	81.82
OWL (Claude sonnet 4)	0.68	1.36	8.29	14.81	20.44	31.65	30.08	45.50	67.39	81.82
Smolagents (Gemini 2.5 Pro)	0.11	0.45	9.01	15.65	18.53	30.91	27.39	45.09	60.00	79.09
Smolagents (Claude sonnet 4)	0.91	0.91	5.06	8.94	14.49	22.68	21.60	33.83	62.95	74.09
Smolagents (GPT-5)	0.45	0.45	8.18	14.27	20.26	30.66	31.83	44.41	66.48	80.00
WebSailor (Gemini 2.5 Pro)	1.25	2.73	12.51	20.49	25.29	39.11	34.41	52.69	70.57	81.36
WebSailor (Claude Sonnet 4)	**2.39**	**3.64**	**16.88**	**24.26**	**32.90**	**42.35**	42.01	**54.01**	70.91	80.90
WebSailor (GPT-5)	0.34	1.36	10.97	16.17	25.96	35.65	37.18	49.48	74.32	**85.00**

Table 2 | Main results on our proposed DeepWideSearch benchmark.

for entity identification and extensive internal knowledge for filling result tables without external search. Furthermore, we detail the performance of baselines from depth and width metrics as below.

Depth Metrics Our analysis reveals that agent systems generally enhance the deep search capabilities of base LLMs, as evidenced by consistent improvements in Core Entity Accuracy (CE Acc.). For example, the CE Acc. (Avg@4) of GPT-5 increases from 58.41% (base LLM) to 74.32% when integrated into WebSailor, representing a +15.91 percentage point gain. Similarly, Claude Sonnet 4 improves from 57.95% to 70.91% under WebSailor, demonstrating the effectiveness of iterative tool calls and multi-step reasoning in complex information retrieval. However, Gemini 2.5 Pro represents a notable exception to this trend. Upon close inspection of generated outputs, we find that Gemini 2.5 Pro in agent systems frequently fails due to three critical issues: (a) producing invalid markdown-formatted tables; (b) executing incorrect tool call APIs; and (c) incomplete task solving due to inference errors, occurring in 24.24% of cases on average—substantially higher than GPT-5 (16.36%) and Claude Sonnet 4 (17.80%). This suggests that Gemini 2.5 Pro's output formatting behavior becomes brittle when subjected to multi-step tool orchestration. Critically, while agent systems improve core entity identification, they fail to consistently enhance column-level precision. For instance, the Column F1 (Avg@4) of Claude Sonnet 4 model declines from 32.63% (base LLM) to 30.08% in OWL and 21.60% in Smolagents. This pattern highlights a fundamental limitation: even when agents successfully identify core entities through multi-hop reasoning, current agent architectures cannot reliably collect complete entities, with their effectiveness often falling below the usage of internal knowledge in base LLMs.

Width Metrics When evaluating width metrics that measure comprehensive information collection, we observe that most agent frameworks do not significantly improve the base LLMs' wide search capabilities. Only three combinations demonstrate consistent improvements across all width metrics:

OWL (Claude Sonnet 4), WebSailor (Claude Sonnet 4), and WebSailor (GPT-5). The remaining agents show substantial performance degradation compared to their counterpart base LLMs. Beyond the issues specific to Gemini 2.5 Pro that described above, the Smolagents framework also consistently underperforms across nearly all metrics. Our investigation reveals that Smolagents employs minimal reasoning before tool calls, which restricts the effectiveness of subsequent tool calls. This architectural constraint prevents Smolagents from formulating precise search queries, resulting in inadequate information coverage and poor performance on width metrics.

6. Analysis

In this section, we conduct several detailed analysis on Efficiency (Section 6.1), Tool Calls (Section 6.2), Differences in Dataset Construction Methods (Section 6.3), Per-topic Performance (Section 6.4), and Error Analysis (Section 6.5).

6.1. Efficiency Analysis

Compared to deep search or wide search, DeepWideSearch imposes significantly higher computational and operational overhead. As shown in Table 4, even state-of-the-art agents incur substantial resource costs per query. For instance, OWL (GPT-5) and WebSailor (Claude Sonnet 4) achieve average $2.75 and $1.40 per question — with many queries remaining unresolved despite this high cost. Due to unstable network conditions and tool call errors, agents often require multiple retry attempts to complete tasks such as search, significantly increasing computational overhead—for instance, OWL (GPT-5) incurs an average cost exceeding $6.8 under retry conditions. These results underscore a critical inefficiency in current agent architectures when tackling complex deep and wide information seeking tasks. This suggests that existing systems are not yet scalable for real-world deployment of DeepWideSearch, motivating future work on efficient planning, memory reuse, and adaptive resource allocation.

Table 3 | Average token usage and cost statistics for some agents on DeepWideSearch questions.

Agents	Input Token	Output Token	Cost ($)
OWL (Gemini 2.5 Pro)	65K	2.5K	≈ 0.2
OWL (GPT-5)	1.8M	50K	**≈ 2.75**
Smolagents (Claude Sonnet 4)	224K	2.4K	≈ 2.14
Smolagents (GPT-5)	120K	25K	≈ 0.90
WebSailor (Gemini 2.5 Pro)	65K	2.5K	≈ 0.49
WebSailor (Claude Sonnet 4)	186.2K	3.5K	≈ 1.40
WebSailor (GPT-5)	17.7K	6.2K	≈ 0.36

6.2. Tool Calls Analysis

Table 4 shows the average number of tool calls (Search and Visit tools) per sample across different backbone LLMs in WebSailor. Notably, WebSailor (Claude Sonnet 4) exhibits a significantly higher Search tool calls (23.23) compared to Gemini 2.5 Pro (4.77) and GPT-5 (8.72). This aligns with its superior performance (Table 2), suggesting that scaling the search tool calls improves the performance.

Table 4 | Average tool calls in the WebSailor system.

Agents	Search	Visit
WebSailor (Gemini 2.5 Pro)	4.77	2.68
WebSailor (Claude Sonnet 4)	23.23	4.57
WebSailor (GPT-5)	8.72	5.35

6.3. Differences in Dataset Construction Methods

Table 5 demonstrates the average performance of advanced LLMs (GPT-5, Claude Sonnet 4 and Gemini 2.5 Pro) with their counterpart agent systems. It can be found that the Deep2Wide construction method produces substantially more challenging data than Wide2Deep method. For example, LLMs and agents achieves nearly 0.0% success rate on Deep2Wide (Avg. LLMs: 0.0% Avg@4; Avg. Agents:

Model / System	**Success Rate (%)**		**Row F1 Score (%)**		**Item F1 Score (%)**		**Column F1 (%)**		**Entity Acc. (%)**	
	Avg@4	Pass@4	Avg@4	Max@4	Avg@4	Max@4	Avg@4	Max@4	Avg@4	Pass@4
Wide Search → *DeepWideSearch* (Wide2Deep)										
Avg. LLMs	1.17	2.22	17.23	21.80	38.04	43.95	50.94	59.09	90.12	93.83
Avg. Agents	1.23	2.13	15.55	24.13	33.51	46.98	44.13	60.70	88.36	96.76
Avg. All	1.21	2.15	16.00	23.49	34.75	46.16	45.96	60.26	88.84	95.96
Deep Search → *DeepWideSearch* (Deep2Wide)										
Avg. LLMs	0.0	0.0	2.67	3.92	8.52	13.25	13.67	21.81	31.77	46.27
Avg. Agents	0.15	0.44	3.25	5.99	9.21	16.43	13.75	24.92	33.86	54.56
Avg. All	0.11	0.32	3.09	5.42	9.02	15.56	13.73	24.07	33.29	52.30
Overall										
Avg. LLMs	0.72	1.36	11.60	14.89	26.64	32.09	36.54	44.69	67.58	75.45
Avg. Agents	0.75	1.25	9.36	14.88	20.76	30.60	27.77	40.74	58.05	69.89
Avg. All	0.74	1.28	9.97	14.88	22.36	31.01	30.16	41.82	60.65	71.40

Table 5 | Performance comparison between Deep2Wide and Wide2Deep methods.

0.15% Avg@4), compared to the Wide2Deep (Avg. LLMs: 1.17% Avg@4; Avg. Agents: 1.23% Avg@4). Critically, the overall Entity Accuracy on Deep2Wide is only 33.29% (vs. 88.84% on Wide2Deep). This observation indicates that the synthesized deep sub-question in the Wide2Deep method is easier for LLMs to solve. Nevertheless, the column-F1 of Wide2Deep remains below 51%, indicating that comprehensively collecting entities is still challenging.

6.4. Per-topic Performance Analysis

As shown in Figure 6, we analyze topic-wise performance through bidirectional bar charts evaluating depth metrics (Column-F1, CE Acc.) and width metrics (Item-F1, Row-F1), excluding domains with fewer than 5 samples. Four key patterns emerge: (1) The top-5 most frequent topics (sample count >20) are *Film & Movies*, *Politics*, *Finance*, *Technology*, and *Sports*; (2) *Politics* achieves the highest item- and row-level F1 scores (35% and 19%), indicating wide search are more tractable in this topic, while *Politics* and *Finance* attain the highest column F1 and CE accuracy, suggesting deep search are comparatively easier here; (3) Despite strong depth performance in *Finance*, *Travel*, and *Education* topics, the performance of baselines exhibit substantially lower width metrics on these three topics (e.g., *Travel* 20% item F1 and *Finance* 8% row F1), revealing that strong deep search capability does not guarantee effective wide search capability; and (4) *History* and *Games* consistently underperform across all metrics (*e.g.*, 5% Column-F1 of *History*), establishing them as the most challenging topics. These findings highlight the heterogeneous nature of search complexity across topics.

6.5. Error Analysis

As shown in Tables 2, agent systems might underperform backbone LLMs on DeepWideSearch tasks. Our error analysis reveals four key failure patterns: (1) **Lack of Reflection:** agents often lack effective reflection mechanisms. When encountering wrong trajectories (Figures 13) or tool call errors (Figure 14), they prematurely conclude the task is unsolvable and output empty tables rather than analyzing failure causes and exploring alternative paths; (2) **Overreliance on Internal Knowledge:** agents frequently overrely on internal knowledge. Even when correctly identifying core entities (Figure 15), they often generate tables solely using their internal parametric knowledge rather than performing proper web queries, resulting in outdated or inaccurate information due to limited training data scope; (3) **Insufficient Retrieval:** information retrieval is often insufficient. For example, despite

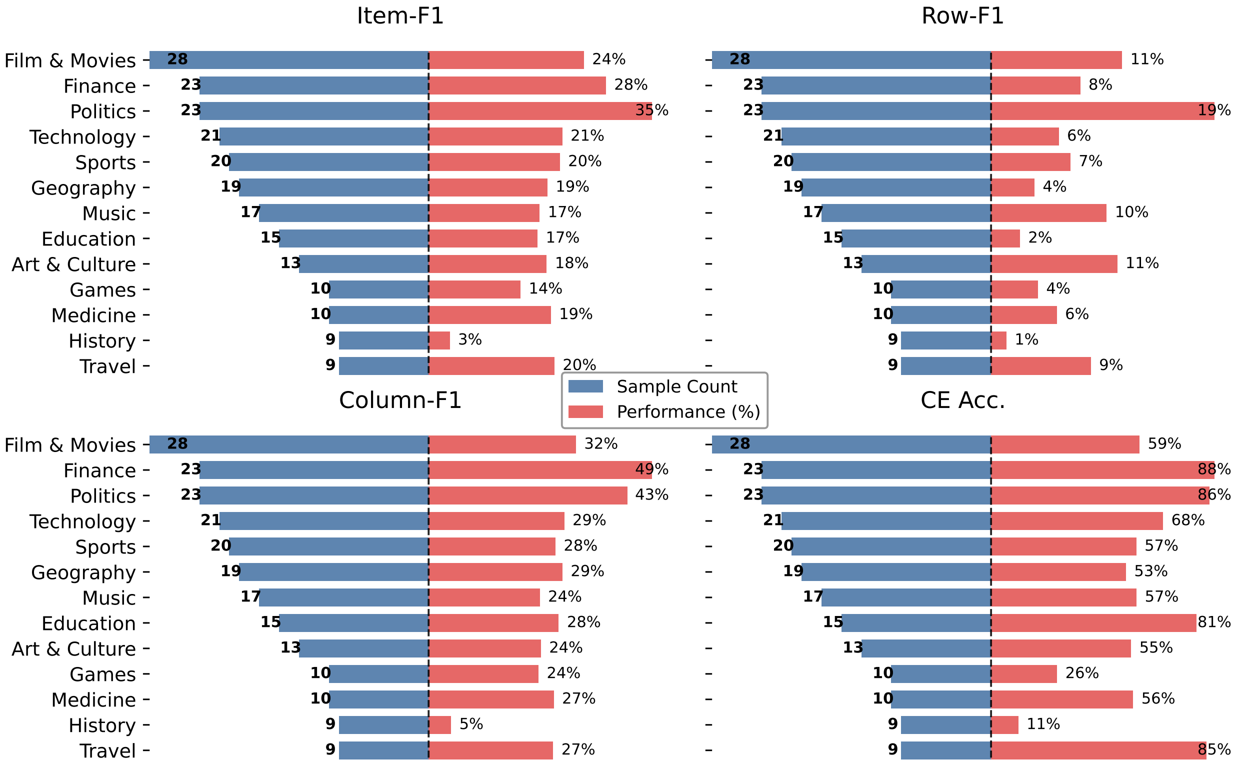


Figure 6 | Per-topic analysis on two depth metrics (Column F1 and CE Acc.) and two width metrics (Item F1 and Row F1).

identifying relevant pages (Figure 17), agents frequently fail to properly access complete context through visit operations, leading to significant information omissions. Even when visit operations are executed correctly, summarized webpage data may still miss critical details. This limitation motivates the design of a question-aware, customized webpage summarization process in agent systems; and (4) **Context Overflow:** context overflow presents a fundamental challenge. Deep wide search requires extensive multi-step reasoning and numerous search tool calls, significantly expanding context length (Figure 16). This issue occurred in 24.96% of cases, exceeding the context management capabilities of current agent architectures; In summary, these four error patterns highlight that current agents face substantial limitations when addressing the challenges of depth and breadth in complex information-seeking tasks. Addressing these limitations requires specialized architecture for deep wide search scenarios.

7. Conclusion

This paper addresses the critical gap in information-seeking agent evaluation by introducing DeepWideSearch benchmark, the first benchmark designed to simultaneously assess deep reasoning and wide-scale information collection. Our experiments demonstrate that state-of-the-art agents achieve only 2.39% average success rate on this challenging benchmark, revealing fundamental limitations for current agents. These results underscore the combinatorial complexity of deep and wide search as a key frontier to guide future research toward more capable information-seeking agents.

8. Limitations and Future Work

Despite our established DeepWideSearch benchmark, there are three key limitations remain to be addressed in the future work: (1) As shown in Table 5, the Wide2Deep construction method produces significantly easier questions than Deep2Wide, as evidenced by the substantially higher CE Accuracy. We will iteratively refine sub-questions to increase question complexity while maintaining natural language quality; (2) Our current dataset exhibits slight differences with real-world deep and wide search questions in terms of solution paths (Cases in Appendix B). In future work, we will iteratively refine the DeepWideSearch dataset to better align with real-world applications; and (3) Our dataset construction relies heavily on human annotation, limiting scalability. Future work should explore automated data generation techniques and develop reference-free evaluation metrics that avoid complex, human-verified tabular answers, enabling efficient dataset expansion and model optimization across diverse domains.

References

J. Achiam, S. Adler, S. Agarwal, L. Ahmad, I. Akkaya, F. L. Aleman, D. Almeida, J. Altenschmidt, S. Altman, S. Anadkat, et al. Gpt-4 technical report. *arXiv preprint arXiv:2303.08774*, 2023.

S. Chen, P. Moreira, Y. Xiao, S. Schmidgall, J. Warner, H. Aerts, T. Hartvigsen, J. Gallifant, and D. S. Bitterman. Medbrowsecomp: Benchmarking medical deep research and computer use, 2025. URL https://arxiv.org/abs/2505.14963.

M. Du, B. Xu, C. Zhu, X. Wang, and Z. Mao. Deepresearch bench: A comprehensive benchmark for deep research agents, 2025. URL https://arxiv.org/abs/2506.11763.

Y. Fan, K. Zhang, H. Zhou, Y. Zuo, Y. Chen, Y. Fu, X. Long, X. Zhu, C. Jiang, Y. Zhang, L. Kang, G. Chen, C. Huang, Z. He, B. Wang, L. Bai, N. Ding, and B. Zhou. Ssrl: Self-search reinforcement learning, 2025. URL https://arxiv.org/abs/2508.10874.

T. Fang, H. Zhang, Z. Zhang, K. Ma, W. Yu, H. Mi, and D. Yu. Webevolver: Enhancing web agent self-improvement with coevolving world model, 2025a. URL https://arxiv.org/abs/2504.21024.

T. Fang, Z. Zhang, X. Wang, R. Wang, C. Qin, Y. Wan, J.-Y. Ma, C. Zhang, J. Chen, X. Li, H. Zhang, H. Mi, and D. Yu. Cognitive kernel-pro: A framework for deep research agents and agent foundation models training, 2025b. URL https://arxiv.org/abs/2508.00414.

H.-a. Gao, J. Geng, W. Hua, M. Hu, X. Juan, H. Liu, S. Liu, J. Qiu, X. Qi, Y. Wu, et al. A survey of self-evolving agents: On path to artificial super intelligence. *arXiv preprint arXiv:2507.21046*, 2025.

B. Gou, Z. Huang, Y. Ning, Y. Gu, M. Lin, W. Qi, A. Kopanev, B. Yu, B. J. Gutiérrez, Y. Shu, C. H. Song, J. Wu, S. Chen, H. N. Moussa, T. Zhang, J. Xie, Y. Li, T. Xue, Z. Liao, K. Zhang, B. Zheng, Z. Cai, V. Rozgic, M. Ziyadi, H. Sun, and Y. Su. Mind2web 2: Evaluating agentic search with agent-as-a-judge, 2025. URL https://arxiv.org/abs/2506.21506.

D. Guo, D. Yang, H. Zhang, J. Song, R. Zhang, R. Xu, Q. Zhu, S. Ma, P. Wang, X. Bi, et al. Deepseek-r1: Incentivizing reasoning capability in llms via reinforcement learning. *arXiv preprint arXiv:2501.12948*, 2025.

R. Han, Y. Chen, Z. CuiZhu, L. Miculicich, G. Sun, Y. Bi, W. Wen, H. Wan, C. Wen, S. Maître, G. Lee, V. Tirumalashetty, E. Xue, Z. Zhang, S. Haykal, B. Gokturk, T. Pfister, and C.-Y. Lee. Deep researcher with test-time diffusion, 2025. URL https://arxiv.org/abs/2507.16075.

Y. He, G. Huang, P. Feng, Y. Lin, Y. Zhang, H. Li, et al. Pasa: An llm agent for comprehensive academic paper search. *arXiv preprint arXiv:2501.10120*, 2025.

M. Hu, Y. Zhou, W. Fan, Y. Nie, B. Xia, T. Sun, Z. Ye, Z. Jin, Y. Li, Q. Chen, Z. Zhang, Y. Wang, Q. Ye, B. Ghanem, P. Luo, and G. Li. Owl: Optimized workforce learning for general multi-agent assistance in real-world task automation, 2025. URL https://arxiv.org/abs/2505.23885.

M. Joshi, E. Choi, D. Weld, and L. Zettlemoyer. TriviaQA: A large scale distantly supervised challenge dataset for reading comprehension. In R. Barzilay and M.-Y. Kan, editors, *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1601–1611, Vancouver, Canada, July 2017. Association for Computational Linguistics. doi: 10.18653/v1/P17-1147. URL https://aclanthology.org/P17-1147/.

K. Li, Z. Zhang, H. Yin, L. Zhang, L. Ou, J. Wu, W. Yin, B. Li, Z. Tao, X. Wang, et al. Websailor: Navigating super-human reasoning for web agent. *arXiv preprint arXiv:2507.02592*, 2025.

A. Liu, B. Feng, B. Xue, B. Wang, B. Wu, C. Lu, C. Zhao, C. Deng, C. Zhang, C. Ruan, et al. Deepseek-v3 technical report. *arXiv preprint arXiv:2412.19437*, 2024.

Y. Lyu, X. Zhang, L. Yan, M. de Rijke, Z. Ren, and X. Chen. Deepshop: A benchmark for deep research shopping agents, 2025. URL https://arxiv.org/abs/2506.02839.

G. Mialon, C. Fourrier, C. Swift, T. Wolf, Y. LeCun, and T. Scialom. Gaia: a benchmark for general ai assistants, 2023. URL https://arxiv.org/abs/2311.12983.

A. Roucher, A. V. del Moral, T. Wolf, L. von Werra, and E. Kaunismäki. 'smolagents': a smol library to build great agentic systems. https://github.com/huggingface/smolagents, 2025.

X. Shi, Y. Li, Q. Kou, L. Yu, J. Xie, and H. Zhou. Spar: Scholar paper retrieval with llm-based agents for enhanced academic search, 2025. URL https://arxiv.org/abs/2507.15245.

H. Sun, Z. Qiao, J. Guo, X. Fan, Y. Hou, Y. Jiang, P. Xie, Y. Zhang, F. Huang, and J. Zhou. Zerosearch: Incentivize the search capability of llms without searching, 2025. URL https://arxiv.org/abs/2505.04588.

Z. Tao, J. Wu, W. Yin, J. Zhang, B. Li, H. Shen, K. Li, L. Zhang, X. Wang, Y. Jiang, P. Xie, F. Huang, and J. Zhou. Webshaper: Agentically data synthesizing via information-seeking formalization, 2025. URL https://arxiv.org/abs/2507.15061.

K. Team, Y. Bai, Y. Bao, and G. C. et al. Kimi k2: Open agentic intelligence, 2025. URL https://arxiv.org/abs/2507.20534.

X. Wang, B. Wang, D. Lu, J. Yang, T. Xie, J. Wang, J. Deng, X. Guo, Y. Xu, C. H. Wu, Z. Shen, Z. Li, R. Li, X. Li, J. Chen, B. Zheng, P. Li, F. Lei, R. Cao, Y. Fu, D. Shin, M. Shin, J. Hu, Y. Wang, J. Chen, Y. Ye, D. Zhang, D. Du, H. Hu, H. Chen, Z. Zhou, H. Yao, Z. Chen, Q. Gu, Y. Wang, H. Wang, D. Yang, V. Zhong, F. Sung, Y. Charles, Z. Yang, and T. Yu. Opencua: Open foundations for computer-use agents, 2025. URL https://arxiv.org/abs/2508.09123.

J. Wei, Z. Sun, S. Papay, S. McKinney, J. Han, I. Fulford, H. W. Chung, A. T. Passos, W. Fedus, and A. Glaese. Browsecomp: A simple yet challenging benchmark for browsing agents, 2025. URL https://arxiv.org/abs/2504.12516.

R. Wong, J. Wang, J. Zhao, L. Chen, Y. Gao, L. Zhang, X. Zhou, Z. Wang, K. Xiang, G. Zhang, et al. Widesearch: Benchmarking agentic broad info-seeking. *arXiv preprint arXiv:2508.07999*, 2025.

J. Wu, B. Li, R. Fang, W. Yin, L. Zhang, Z. Tao, D. Zhang, Z. Xi, G. Fu, Y. Jiang, P. Xie, F. Huang, and J. Zhou. Webdancer: Towards autonomous information seeking agency, 2025a. URL https://arxiv.org/abs/2505.22648.

J. Wu, W. Yin, Y. Jiang, Z. Wang, Z. Xi, R. Fang, L. Zhang, Y. He, D. Zhou, P. Xie, and F. Huang. Webwalker: Benchmarking llms in web traversal, 2025b. URL https://arxiv.org/abs/2501.07572.

Y. Xi, J. Lin, Y. Xiao, Z. Zhou, R. Shan, T. Gao, J. Zhu, W. Liu, Y. Yu, and W. Zhang. A survey of llm-based deep search agents: Paradigm, optimization, evaluation, and challenges. *arXiv preprint arXiv:2508.05668*, 2025.

Y. Xia, J. Fan, W. Chen, S. Yan, X. Cong, Z. Zhang, Y. Lu, Y. Lin, Z. Liu, and M. Sun. AgentRM: Enhancing agent generalization with reward modeling. In W. Che, J. Nabende, E. Shutova, and M. T. Pilehvar, editors, *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 19277–19290, Vienna, Austria, July 2025. Association for Computational Linguistics. ISBN 979-8-89176-251-0. doi: 10.18653/v1/2025.acl-long.945. URL https://aclanthology.org/2025.acl-long.945/.

W. Xu, K. Mei, H. Gao, J. Tan, Z. Liang, and Y. Zhang. A-mem: Agentic memory for llm agents, 2025. URL https://arxiv.org/abs/2502.12110.

A. Yang, A. Li, and B. Y. et al. Qwen3 technical report, 2025. URL https://arxiv.org/abs/2505.09388.

Z. Yang, P. Qi, S. Zhang, Y. Bengio, W. W. Cohen, R. Salakhutdinov, and C. D. Manning. Hotpotqa: A dataset for diverse, explainable multi-hop question answering, 2018. URL https://arxiv.org/abs/1809.09600.

W. Zhang, C. Cui, Y. Zhao, R. Hu, Y. Liu, Y. Zhou, and B. An. Agentorchestra: A hierarchical multi-agent framework for general-purpose task solving. *arXiv preprint arXiv:2506.12508*, 2025a.

Z. Zhang, Z. Chen, M. Li, Z. Tu, and X. Li. Rlvmr: Reinforcement learning with verifiable meta-reasoning rewards for robust long-horizon agents, 2025b. URL https://arxiv.org/abs/2507.22844.

H. Zhou, X. Wan, R. Sun, H. Palangi, S. Iqbal, I. Vulić, A. Korhonen, and S. Arık. Multi-agent design: Optimizing agents with better prompts and topologies, 2025a. URL https://arxiv.org/abs/2502.02533.

P. Zhou, B. Leon, X. Ying, C. Zhang, Y. Shao, Q. Ye, D. Chong, Z. Jin, C. Xie, M. Cao, Y. Gu, S. Hong, J. Ren, J. Chen, C. Liu, and Y. Hua. Browsecomp-zh: Benchmarking web browsing ability of large language models in chinese, 2025b. URL https://arxiv.org/abs/2504.19314.

M. Zhuge, C. Zhao, D. R. Ashley, W. Wang, D. Khizbullin, Y. Xiong, Z. Liu, E. Chang, R. Krishnamoorthi, Y. Tian, Y. Shi, V. Chandra, and J. Schmidhuber. Agent-as-a-judge: Evaluating agents with agents, 2025. URL https://openreview.net/forum?id=DeVm3YUnpj.

A. Details of Datasets

The table volume in Table 1 represents the number of the searched information in the DeepWideSearch questions, which is defined as the product of rows and columns of the table. The average steps of the search entities is counted as the number of the reasoning steps and tool calls. Specifically, the average steps of GAIA is counted by the reference trajectories in the dataset, and the average steps of WideSearch is annotated by our three human raters. Besides, Figure 7 and Figure 8 present two cases in our proposed DeepWideSearch dataset.

Case Deep2Wide_62 Instance (Core Entity is Nobel Proze)

DeepWide Question: Soviet physicist A received his Ph.D. at the age of 27 under the supervision of the renowned Soviet physicist B. Physicist B was awarded the Nobel Prize at the age of 54 and passed away six years later. In 2023, physicist A received a prestigious international prize in physics. Please provide the names, specific award titles, dates of birth (formatted as "Mon DD, yyyy"; if the exact date is unknown, use "-, yyyy"), and nationalities of the other scientists who received awards in the same year as A. Present the results in a single Markdown table with the following columns in order: Name, Award, Date of Birth, Nationality. All cells must be filled according to the column requirements; do not omit any information arbitrarily. The table must be output entirely in Chinese, and the final output should strictly follow the format: ```markdowntable content```.

Ground-truth Table of Deep2Wide_62 Instance

Name	Award	Date of Birth	Nationality
Peter Agre	Chem	Jan 30, 1949	United States
Roderick MacKinnon	Chem	Feb 19, 1956	United States
Clive Granger	Econ	Sep 04, 1934	United Kingdom
Robert Engle	Econ	Nov 10, 1942	United States
Alexei Abrikosov	Phys	Jun 25, 1928	Russia/United States
Vitaly Ginzburg	Phys	Oct 04, 1916	Russia/United States
Anthony Leggett	Phys	Mar 26, 1938	United Kingdom/United States
J. M. Coetzee	Lit	Feb 09, 1940	South Africa
Shirin Ebadi	Peace	Jun 21, 1947	Iran
Paul Lauterbur	Med	May 06, 1929	United States
Peter Mansfield	Med	Oct 09, 1933	United Kingdom

Figure 7 | One case in DeepWideSearch dataset.

Case Deep2Wide_26 Instance (Core Entity is Chen Yixun)

DeepWide Question: A pop song performed by a well-known Chinese singer, in which the song title appears ten times in the lyrics, and the first two words of the album title start with the same letter. The lyricist once served as a judge on a music variety show and later appeared on a local TV program alongside another famous lyricist. Could you please list the TV dramas the singer has acted in, and summarize their titles, directors, and main cast? Present the results in a Markdown table with the columns in the following order: Name, Director, Main Cast. Output only the result in the format: ```markdowntable content```.

Ground-truth Table of Deep2Wide_26 Instance

Name	**Director**	**Main Cast**
Brief Marriage	Chen Zhifa	Eason Chan, Cecilia Yip
Toward Happiness	Lin Jianzhong	Nicky Wu, Tang Yuhong
Bloody Marriage	Jin Ge	Gu Shaohua
Sapphire Night Sky	Lai Hoi Shan	Charlene Choi, Eason Chan
Let's Cheer Together	Zhu Ruibin	Steven Cheung, Kenny Kwan
Triumph in the Skies	Poon Ka Tak	Anthony Wong, Myolie Wu
When Four-Leaf Clover Meets the Sword Tip	Zhang Yijie	Lai Lok Yi, Li Rilang
Pet Love	Cheng Kei Sing	Louis Koo, Sammi Cheng
Midnight Express	Ono Tetsujiro Takemura Kentaro	Takao Osawa, Nanako Matsushima

Figure 8 | One case in DeepWideSearch dataset.

B. Differences between Our Dataset and Real-world Questions

Two Deep and Wide Questions

Our DeepWideSearch Question: Identify an artist who studied at both the Central Academy of Fine Arts in China and the Kunstakademie Düsseldorf in Germany, and who pursued further studies in Germany. During his time in Germany, he studied under three renowned artists, one of whom held the record for the highest auction price ever achieved by a living artist in 2012. Gather information about this artist's solo exhibitions, including exhibition title, venue, city, and exhibition dates.
Solution Path: first identify the artist (deep/wide search), then perform wide search for information collection (wide search).

Real-world Deep and Wide Question: Please help me identify emerging local merchants from Thailand and Vietnam that operate pet supplies categories on e-commerce platforms such as Lazada and Shopee, and whose store GMV growth rate exceeded 50% in the first half of 2025. (Emerging merchants are defined as those that opened their stores after 2023.) I need the following information for each qualifying store: store URL, company legal entity details, business contact email, and the founder's LinkedIn profile.
Solution Path: first perform wide search to list candidate stores (wide search), then perform deep search to collect their information (GMV) for verification (deep search).

Figure 9 | Two cases of the deep and wide search questions.

Figure 9 illustrates two representative deep and wide search questions: the first is an example from our constructed DeepWideSearch dataset, and the second is drawn from a real-world e-commerce scenario. While our dataset captures the essential characteristics of deep and wide search, the primary difference from real-world settings lies in the solution path. In our dataset, the process emphasizes first performing a deep search to gather critical information, followed by a wide search to expand relevant attributes. In contrast, real-world tasks often begin with a wide search to collect a large pool of candidates, followed by a deep search over each candidate for verification. Nevertheless, it is important to emphasize that despite this procedural difference, our dataset still exhibits the traits of deep and wide search. Specifically, during the initial deep search phase, the model also need to list and reason over a set of candidates, systematically applying deep verification to determine which candidates satisfy the problem constraints and thereby identify the correct target entity. Consequently, even this first-stage deep search inherently incorporates the characteristic of the wide search.

C. Prompts for DeepWideSearch Data Construction

This section presents three prompts for Wide2Deep method: (1) Core Entity Extraction Prompt in Figure 10; (2) Deep Sub-Question Synthesis Prompt in Figure 11; (3) Question Fusion in Figure 12.

# **You are a professional entity extraction expert tasked with identifying the most critical entity from a given text query.**

# **Classification Requirements:**

- The entity must be a specific, concrete entity object mentioned within the text query.
- A query may contain multiple entities, but prioritize outputting the single most central entity. If no single core entity can be determined, output multiple entities.
- Output ONLY JSON-formatted data containing the entities, with no additional content, explanations, or numbering (to ensure direct parseability). Please directly output the entity, without any explanation.

# **Reference Few-shot Examples**

* Input:

I absolutely love Jay Chou. Please find all songs released by Jay Chou between January 2004 and September 2010 (including January 2004 and September 2010). Include song details: title, lyricist, composer, release date, album, and duration.

Notes:

1. I want only Jay Chou's original vocals (collaborations allowed), excluding instrumental tracks.
2. Format release dates as yyyy/mm/dd; duration as x minutes x seconds (e.g., 3 minutes 5 seconds).
3. Include only songs released in China.
4. Exclude live versions and demos.
5. Include only songs from Jay Chou's studio albums (exclude singles, single albums, film/TV soundtracks, EPs).

* Output: Jay Chou

* Input:

I want to register for the 2026 postgraduate entrance exam. Please check the (retest) for the Journalism and Communication program (full-time professional master's) at Chinese universities in Region A with 211 Project status or higher for 2025 (total score only).

* Output: Journalism and Communication program

* Input:

Using statistics from the Stockholm International Peace Research Institute (SIPRI), list the specific military expenditures (in billion USD without decimals, e.g., 9000 billion USD), GDP (in trillion USD to two decimal places, e.g., 30.21 trillion USD), global military expenditure ranking, head of state (actual leader), and defense minister for the US, Russia, Germany, India, and Japan for each year from 2019 to 2024 (inclusive).

* Output: Stockholm

* Input:

I'm running out of books to read. Could you compile a ranked list of the top 10 books from Douban Reading's annual (overall) for 2022-2024 (inclusive), plus the top 10 bestsellers and highest-rated books from Dangdang.com each year? Include authors' names.

* Output: Douban Reading, Dangdang.com

* Input:

Please compile a table listing the "CNN Hero of the Year" for every year the award was actually presented, from its first introduction through 2024 (including 2024), along with relevant details for each honoree.

* Output: CNN Hero of the Year

# **Perform entity extraction for the following query:**

{question}

Figure 10 | The prompt of core entitiy extraction in Wide2Deep method.

Please help me gather all available information about “{entity}”, and based on the collected information, synthesize a multi-hop query that meets the following requirements:
(1) The answer to the query must be exactly “{entity}” only—no other answers or ambiguities allowed;
(2) The information about “{entity}” included in the query should not be overly specific (e.g., exact dates, locations, awards, or distinctive features), so that one cannot directly find “entity” by searching those fragments online;
(3) A human answering the query must perform multiple search steps and gather information from at least three distinct, non-repeating URLs to logically infer “{entity}”;
(4) The query must be concise. Avoid constructing long queries by piling up excessive features. Include only 2–3 pieces of information about {entity}. Focus on the ambiguity and reliability of these features, rather than the quantity;
(5) Consider temporal validity—the answer to the query must remain stable over time and not change with time;
(6) After generating the query, you must conduct an additional verification process via search engines: extract 3–5 simplified search queries from the synthesized query (each reflecting one key feature or phrase from the original query; keep them short and focused on core keywords). Analyze the search results to ensure that “{entity}” cannot be directly found in a single step. **If “{entity}” can be found directly through any of these 3–5 searches, the synthesized query is too simple and does not meet requirements—please repeat the entire process until none of the derived searches can directly reveal “{entity}”.**

# Output Format: Place all generated query, reference URLs, and reasoning in the standard JSON structure

Figure 11 | The prompt of deep sub-question synthesis in Wide2Deep method.

D. Error Cases in DeepWideSearch

This section provides the four kinds of representative errors of agents: (1) Lack of Reflection (Figure 13 and Figure 14); (2) Overreliance on Internal Knowledge (Figure 15); (3) Context Overflow (Figure 16); and (4) Insufficient Retrieval (Figure 17).

You are a senior question synthesis expert responsible for integrating a complex question about an entity into an original query containing that entity, thereby constructing a more challenging composite query. Please strictly follow the rules below:
1. **Task Objective**:
- Replace the entity in the original query with the corresponding complex question.
- The generated query must require users to first resolve the complex question to obtain the entity information, then proceed with the information retrieval steps in the original query.
- The final synthesized query must be grammatically correct and logically coherent.
- The complex question is in Chinese, but the original query may be in either Chinese or English: if the original query is in English, you must translate the complex question into accurate and equivalent English and integrate it into the query (without adding or omitting any information from the original Chinese question, ensuring full translation fidelity); if the original query is in Chinese, the final synthesized query must also be in Chinese.

2. **Input Specifications**:
- Original query (query): {query}
- Entity (entities): entities (for internal reference only; **must not appear in the output**)
- Complex question (question): A complex query question to find entities: question

3. **Synthesis Rules**:
- Replace the entity in the original query with the descriptive text of the complex question.
- The complex question must be transformed into a noun phrase (remove the question mark and rephrase it as a descriptive clause).
- **Do not** reveal any entity information (specific names from “entities” must not appear).
- Maintain professional translation quality and avoid awkward or unnatural phrasing.

Output only the new synthesized query that integrates the complex question with the original query, without any explanations or additional content

Figure 12 | The prompt of deep and wide question fusion in Wide2Deep method.

Error: Lack of Reflection

Input of Deep2Wide_62 Instance

DeepWide Question: Soviet physicist A received his Ph.D. at the age of 27 under the supervision of the renowned Soviet physicist B. Physicist B was awarded the Nobel Prize at the age of 54 and passed away six years later. In 2023, physicist A received a prestigious international prize in physics. Please provide the names, specific award titles, dates of birth (formatted as "Mon DD, yyyy"; if the exact date is unknown, use "-, yyyy"), and nationalities of the other scientists who received awards in the same year as A. Present the results in a single Markdown table with the following columns in order: Name, Award, Date of Birth, Nationality. All cells must be filled according to the column requirements; do not omit any information arbitrarily. The table must be output entirely in Chinese, and the final output should strictly follow the format: “““markdowntable content““.
Core Entity: Nobel Prize

Multi-turn Converation of WebSailor (GPT-5)

...

~~~
{
    "role": "assistant",
    "content": "We first uniquely identify physicist B as Lev Landau ... An exhaustive review of these major 2023 physics prizes reveals no recipient who was a student of Lev Landau. Furthermore, ... Consequently, under the conditions specified in the problem (A being a student of Landau and receiving a major international physics award in 2023), no matching individual can be identified in authoritative sources ... Given the inability to verify the identity of A or the specific award received, it is impossible to reliably list. Although the problem requires a table output, academic integrity precludes the fabrication of data. Therefore, an empty table is provided, with placeholders (“-”) indicating the absence of verified entries. Dates are formatted as “Mon DD, yyyy” per the instructions.
```markdown
| Name | Award | Date of Birth | Nationality |
|—|—-|———|——-|
| - | - | - | - |
```"
}
~~~

Figure 13 | Lack of Reflection when dive into the wrong trajectory.

Error: Lack of Reflection

Input of Wide2Deep_056 Instance

DeepWide Question: I am particularly interested in a discipline that is grounded in foundational beliefs requiring no empirical verification, derives deterministic conclusions through rigorous transformation rules, and is entirely detached from the physical world. Please help me compile a list of institutions rated A+ in this discipline according to the Ministry of Education's fourth round of disciplinary evaluations, along with specific information about academicians in this discipline at those institutions. Requirements: (1) Only include academicians who had already received the title of academician during their tenure at the institution's school/faculty of this discipline; (2) Only include those who held positions prior to 2025 (excluding 2025); (3) Exclude any academicians who passed away on or before June 2025. Please present the compiled data in a single Markdown table—do not split it into multiple tables. Every cell must be filled according to the column requirements; do not omit any information arbitrarily. Use Chinese for the output content. The column headers, in order, are: Institution Name, Academician Name, Year of Birth, Awards Received. Note: If any field cannot be found via online searches, do not fabricate information; instead, enter "NA" in the corresponding cell. For awards, only check for the Chern Shiing-Shen Mathematics Award and the Hua Luogeng Mathematics Award; if neither applies, output "NA". Do not ask me any questions—only output the result in the following format: ```markdowndata content```.
Core Entity: Mathematic

Multi-turn Converation of WebSailor (GPT-5)

...

~~~
{
    "role": "assistant",
    "content": "<think>First, confirm that the discipline described by the user is “Mathematics,” characterized by its axiomatic foundations, deductive reasoning, and independence from the physical world. Next, identify the institutions awarded an “A+” rating in Mathematics in the Ministry of Education’s Fourth Round of Discipline Evaluations (released in 2016) ... Given the inability to conduct correct online searches in this context—any attempt to populate the requested data would risk inaccuracy ... After careful consideration, the safest and most ethically consistent course—adhering strictly to the “no speculation” principle—is to provide a table header without any data rows. ...:
```markdown
| Institution Name | Academician Name | Year of Birth | Award Received |
``` }
~~~

Figure 14 | Lack of reflection when tool calls are wrong.

Error: Overreliance on Internal Knowledge

Input of Wide2Deep_059 Instance

DeepWide Question: Which company, headquartered in a Silicon Valley city named after an Italian saint, has committed to achieving carbon neutrality by 2030 and employs more than 160,000 individuals? I am currently investigating the evolutionary trajectory of this company's products and require a compilation of its primary flagship smartphone product lines launched in the U.S. market from January 9, 2007, to December 31, 2024. For each generation, the following data must be recorded: product name, launch year, storage capacity (e.g., 128 GB), official launch price in U.S. dollars under carrier contract (prices corresponding to each storage option listed in a single row, separated by "/", e.g., capacity: "4 GB/8 GB", price: "xxx$/xxx$", with strict one-to-one alignment between capacities and prices), and the most significant new technological feature or functional innovation introduced with that generation. The scope is strictly limited to the standard annual flagship product line released each year in September or October. Each product line should occupy exactly one row in the table. If the launch price or core new feature cannot be verified, the respective fields should be left blank. Please present the compiled data in a single Markdown table—do not split it into multiple tables. Every cell must conform to the specified column requirements; no information should be omitted without justification. The column headers, in sequence, are: Product Name, Launch Year, Storage Capacity, Launch Price by Capacity (USD), Core New Technology / Feature. Do not pose any questions—output only the result in the following format: ```markdowndata content```
Core Entity: Apple Inc.

Multi-turn Converation of WebSailor (GPT-5)

...

~~~
{
    "role": "assistant",
    "content": <think>I first determine that the company described by the user is Apple Inc ... I
  then need to compile data on the regular flagship iPhone product lines launched between 2007 and
  2024, specifically those released in September or October each year. To ensure pricing accuracy,
  I will only include two-year contract prices when explicitly stated in official Apple press
  releases ... The final output is presented in a single Markdown table, with storage capacities and
  corresponding prices listed in matching order, separated by slashes.</think><answer>```markdown
  | Product Name | Launch Year | Storage Capacity | Launch Price by Capacity (USD) | Core New
  Technology / Feature |
  | – | – | – | – | – |
  ...
  ```</answer> }
~~~

Figure 15 | Overreliance on the internal knowledge of LLMs.

Error: Context Overflow

Input of Wide2Deep_095 Instance

DeepWide Question: Please help me compile a list of the athlete who won an Olympic gold medal as a key player in the 1980s, was later inducted into the international hall of fame of her sport with 100% voting support, and in the 21st century served as head coach of another country's national team, winning a medal at an Olympic Games hosted by her country of birth. Additionally, compile all results and information from major international competitions during her tenure as head coach of the Chinese Women's National Volleyball Team (note two terms: April 2013 – August 2016; April 2017 – August 2021). Only include competitions in which she served as head coach. The major international competitions include: the Olympic Games, FIVB Volleyball Women's World Championship, FIVB Volleyball Women's World Cup, Volleyball Nations League (VNL), Asian Women's Volleyball Championship, AVC Cup, FIVB World Grand Champions Cup, Montreux Volley Masters, FIVB Volleyball World Grand Prix, and Asian Games. The following information is required for each entry: year, full official name of the competition (e.g., 2016 Rio Olympics), names of Chinese women's volleyball team players who participated (list only names, separated by commas, without positional details), and the final result/ranking of the team. Present all compiled data in a single Markdown table—do not split into multiple tables. Every cell must contain the required information according to the column headers; no omissions are permitted. Use English for all content within the table. Column headers, in order, are: Year, Competition Name, Chinese Women's Volleyball Team Players, Final Result/Ranking. Ensure competition names are complete (e.g., "2016 Rio Olympics"). Do not ask any questions—only output the result in the following format: ```markdowndata content```.
Core Entity: Ping Lang

Multi-turn Converation of WebSailor (GPT-5)

...

~~~
{
    "role": "assistant",
    "content": "You have now reached the maximum context length you can handle. You should stop
 making tool calls and, based on all the information above, think again and provide what you
 consider the most likely answer in the following format:<think>your final thinking</think>
 <answer>your answer</answer> ... ```markdown
 ...
 ```" }
~~~

Figure 16 | Multi-turn tool calls and reasoning leads to the context overflow problem, and agents are interrupted to output the table.

Error: Insufficient Retrieval

Input of Wide2Deep_038 Instance

DeepWide Question: I am researching a type of designated area that complies with specific national standards, is subject to dynamic management by the Ministry of Culture and Tourism, undergoes re-evaluation every three years, covers no less than 5 square kilometers, and currently includes a total of 85 such areas nationwide. Please provide a complete statistical table as of 2025, listing for each designated area its name, the province in which it is located, and its designation year (formatted as, e.g., “2010”). Present the data in a single Markdown table with the following column headers in English: Area Name, Province, Designation Year. Do not split the data into multiple tables, ensure every cell contains the required information without omissions, and output the table entirely in English. Do not ask any questions—only output the result in the format ```markdowndata```.
Core Entity: China National Tourist Resort

Multi-turn Converation of WebSailor (GPT-5)

...

```
{
    "role": "assistant",
    "content": "I need to confirm which type of regional designation corresponds to the user’s
description ... My preliminary judgment is that it refers to China National Tourist Resorts. ...
<tool_call>{"name": "search" ...} ..." }
```

```
{
    "role": "user",
    "content": "<tool_response>
A Google search for ’China National Tourist Resorts 85’ found 10 results:
## Web Results ..." }
```

```
{
    "role": "assistant",
    "content": "Based on the searched results, we need to output the complete list of "National
Tourist Resorts" as of 2025 (a total of 85), indicating the respective province and year of
designation ..." }
```

Figure 17 | Complete information in the webpages are not passed to the agents, leading to the insufficient retrieval error.