OAgents: An Empirical Study of Building Effective Agents

He Zhu*[1], Tianrui Qin*[1], King Zhu[1], Heyuan Huang[1], Yeyi Guan[1,◇], Jinxiang Xia[1,◇]
Yi Yao[1], Hanhao Li[1,◇], Ningning Wang[3], Pai Liu[1,◇], Tianhao Peng[3], Xin Gui[1,◇]
Xiaowan Li[1,◇], Yuhui Liu[3],Yuchen Eleanor Jiang[1], Jun Wang[1], Changwang Zhang[1]
Xiangru Tang[5], Ge Zhang[3], Jian Yang[3], Minghao Liu[4], Xitong Gao[6]
Jiaheng Liu[2,†] and Wangchunshu Zhou[1,†]
[1] OPPO Research Institute
[2] Nanjing University [3] M-A-P [4] 2077.ai [5] Yale University
[6] Shenzhen Institutes of Advanced Technology, Chinese Academy of Sciences
{zhuhe, qintianrui, zhouwangchunshu}@oppo.com, liujiaheng@nju.edu.cn

Abstract

Recently, Agentic AI has become an increasingly popular research field. However, we argue that current agent research practices lack standardization and scientific rigor, making it hard to conduct fair comparisons among methods. As a result, it is still unclear how different design choices in agent frameworks affect effectiveness, and measuring their progress remains challenging. In this work, we conduct a systematic empirical study on *GAIA*benchmark to examine the impact of popular design choices in key agent components in a fair and rigorous manner. We find that the lack of a standard evaluation protocol makes previous works, even open-sourced ones, non-reproducible, with significant variance between random runs. Therefore, we introduce a more robust evaluation protocol to stabilize comparisons. Our study reveals which components and designs are crucial for effective agents, while others are redundant, despite seeming logical. Based on our findings, we build and open-source OAGENTS, a new foundation agent framework that achieves state-of-the-art performance among open-source projects. OAGENTS offers a modular design for various agent components, promoting future research in Agentic AI.

1 Introduction

In recent years, language agents (Significant-Gravitas, 2023; Wu et al., 2023; Roucher et al., 2025; Li et al., 2023; Zhou et al., 2023b; Xie et al., 2023; Zhou et al., 2024) have received significant attention due to their potential in resolving general, complex tasks that traditionally required human intervention. However, despite the surge in the number of research works and open-sourced agent frameworks, current practices in Agentic AI research are far from being rigorous and scientific. Specifically, the current landscape of agent research suffers from a lack of standardized designs and implementation details. Critical components such as planning (Yao et al., 2023; Shinn et al., 2023; Liu et al., 2024), memory (Zhou et al., 2023a; Zhang et al., 2024; Xu et al., 2025), and tool use (Qin et al., 2024; Wang et al., 2024) vary widely across different papers and frameworks, making it difficult to attribute performance improvements to specific innovations. Compounding this issue, reported results are often hard to reproduce due to inconsistent evaluation settings or undisclosed framework configurations (Hu et al., 2025; at Ant Group, 2025). This fragmentation undermines the scientific rigor of the field, as findings cannot be reliably compared or built upon.

Take the widely researched GAIA Benchmark (Mialon et al., 2023) as an example. Despite the organizers provide a public leaderboard with evaluation code and a number of papers and projects being open-sourced, it is still very hard, if not impossible, for other researchers to reproduce their results because a number of inconspicuous factors are not standardized, including the implementation details of tools and prompts, as well as details in the evaluation protocol such as how many runs are performed, how errors and failures are handled, and how different results are ensembled or aggregated. These factors often lead to a large impact on the overall performance, sometimes the impact is even larger than some new architecture innovations in new research papers. However, they are generally not mentioned in the technical reports of different agent frameworks and are not even included in their open-sourced codebases. Moreover, the engineering design and details in different agent research papers and codebases are so large that it makes it impossible to conduct apples-to-apples comparisons on specific technical designs. This

* Equal Contribution.
◇ Work done during internship at OPPO.
† Correspondence to: Jiaheng Liu and Wangchunshu Zhou.

Findings of the Association for Computational Linguistics: EMNLP 2025, pages 13354–13369
November 4-9, 2025

makes it very hard for the research community on agentic AI to properly conduct scientific research instead of digging into tricks on engineering details and evaluation protocols. As a result, despite a lot of agent research papers being released and the numbers on public benchmarks keeping increasing, the best practices on building effective agents are still very obscure.

In this work, to promote truly scientific research on agentic AI and provide researchers with a clear understanding of key factors in building effective agents, we conduct a systematic empirical study on the GAIA benchmark to sort out the core design choices in current agent research, analyze their impact on performance, and report practical tips for improving experimental stability and reproducibility. Specifically, we: (1) carefully implement and compare different designs on key agent components including planning, tool use, memory, test-time scaling strategies, etc., (2) systematically investigate the impacts of different LLM backbones and their combinations; (3) thoroughly analyze different practices for evaluation and provide a more robust evaluation protocol.

Based on the empirical study, we implement and release OAGENTS, a language agent framework that achieves state-of-the-art performance among open-sourced agent frameworks on the GAIA benchmark. More importantly, OAGENTS supports modularized integration of almost all critical designs and features in critical components of language agents, including: (1) different agentic planning mechanisms including static and dynamic workflow designs; (2) a complete tool box including web search with different search sources, browsing tools and web crawlers, parsing tools compatible with more document types. (3) different design of the agentic memory module; (4) test-time scaling strategies for agents including different search algorithms and reflection/self-refine mechanisms. Hopefully, OAGENTS will facilitate scientific research on language agents by promoting apple-to-apple comparisons and standardizing evaluation protocols.

In summary, our main contributions are as follows.

(1) We present a comprehensive agent framework - OAGENTS. OAGENTS encompass periodically revised plan generation, fine-grained task decomposition & simultaneous execution, optimization of multi-source web browsing, enhanced document parsing, and adaptive memory mechanisms that collectively enhance performance across various tasks, ranking 1st among open-source agent frameworks on the GAIA benchmark.

(2) We conduct a systematic empirical study and performance analyzes based on the OAGENTS framework, offering principles to decompose, analyze and optimize agent designs, uncovering optimal architectural choices and key factors influencing experimental stability.

(3) We introduce practical techniques for reducing experimental variance, including optimization of inference parameters and majority voting strategies, enabling a more reliable and consistent evaluation of agent performance.

2 Related Work

Existing work primarily develops agent frameworks along two dimensions: Role specialization paradigms construct collaborative networks through differentiated tool allocation (e.g., AutoGPT (Significant-Gravitas, 2023), AutoGen (Wu et al., 2023), and Camel (Li et al., 2023)) or functional partitioning(e.g., Barcelona2, Omne, AgentIM). Smolagents (Roucher et al., 2025) combines the ReAct (Yao et al., 2023) and Code Act (Wang et al., 2024) architectures to build a multi-functional agents hierarchy to perform multiple rounds of interactions and actions in code to accomplish complex tasks. Magentic-One (Fourney et al., 2024) achieves efficient processing of vision-language tasks by decoupling perception (Yang et al., 2023a,b), planning (Song et al., 2023; Tordesillas and How, 2021), and execution modules (Qin et al., 2024; Wang et al., 2024). Dynamic orchestration mechanisms include Trase-Agent (Trase, 2024) which proposes task reallocation strategies based on real-time feedback, while TapeAgents (Bahdanau et al., 2024) employs an asynchronous communication framework to enhance system resilience. Experimental evidence suggests that stable sub-agent environment interactions provide greater task success rates than complex orchestration algorithms. AutoAgent (Tang et al., 2025) enables intelligent task execution and personalized agent creation without coding through the core components such as natural language-driven multi-agent coordination, customizable workflows, and self-managing file systems. Hybrid architecture exploration is exemplified by h2oGPTe-Agent (H2O.ai, 2024), which transfers single agent

These are closed-source framworks.

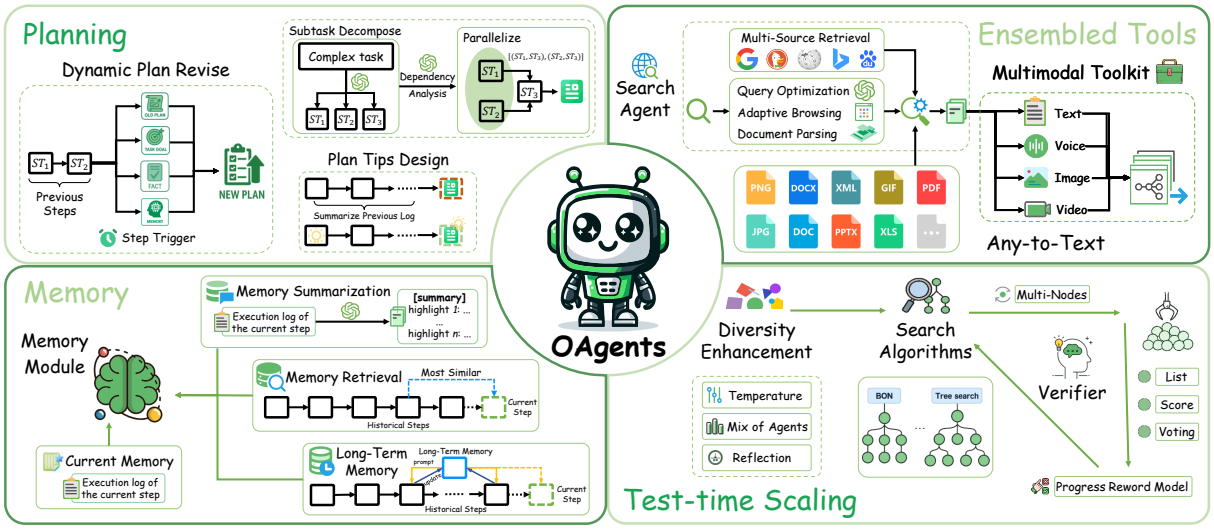

Figure 1: The key components of the OAGENTS framework, including planning, memory, tools, and test-time scaling.

optimization techniques to multi-agent scenarios, achieving over 70% accuracy in code generation tasks. However, it still encounters significant bottlenecks in cross-modal tasks.

3 Building Effective Agents

Table 1: Performance of various agent frameworks on the *GAIA* benchmark.

Framework	Model Family	Avg.	Level 1	Level 2	Level 3
Agentic Model					
Search-o1-32B	-	39.8	53.8	34.6	16.7
WebThinker-32B-RL	-	48.5	56.4	50.0	16.7
Closed-source Agent Frameworks					
Langfun Agent	Claude-3-7 *etc.*	71.52	83.02	68.60	57.69
TraseAgent	Claude *etc.*	70.30	83.02	69.77	46.15
Deep Research	Unknown	67.36	74.29	69.06	47.60
h2oGPTe	Claude-3.5	63.64	67.92	67.44	42.31
Desearch	GPT-4o	56.97	71.70	58.14	23.08
Open-Source Agent Frameworks					
OWL - Workforce	Claude-3-7 *etc.*	69.09	84.91	67.44	42.31
OWL-Roleplaying	4o & o3-mini *etc.*	58.18	81.14	54.65	23.08
TapeAgents	Claude-3-7 *etc.*	55.76	71.70	53.49	30.77
AutoAgent	Claude-3-5 *etc.*	55.15	71.70	53.40	26.92
Open Deep Research	OpenAI o1	55.15	67.92	53.49	34.62
Smolagents	Openai o1 *etc.*	53.33	62.26	54.65	30.77
Magnetic-1	OpenAI o1 *etc.*	46.06	56.60	46.51	23.08
FRIDAY	GPT-4 turbo	34.55	45.28	34.88	11.54
OAGENTS	Claude-3-7 *etc.*	66.67	77.36	66.28	46.15
OAGENTS-Pass@3	Claude-3-7 *etc.*	**73.93**	**83.02**	**74.42**	**53.85**

We present a dual-axis analytical paradigm for architecting cognitive agents in open-world environments, focusing on two orthogonal evaluation dimensions: *factual acquisition capacity (FAC)* and *logical reasoning fidelity (LRF)*. The FAC axis quantifies an agent's proficiency in assimilating and updating domain-specific knowledge from dynamic information streams, while the LRF axis measures its capability to maintain rigorous causal relationships and deduction chains during complex problem-solving. Through systematic examination of these complementary dimensions, we establish methodological guidelines for *1) Enhancing environmental perception through adaptive knowledge integration* and *2) Ensuring decision-making robustness via verifiable inference processes*. This bifocal approach addresses the fundamental challenges of balancing empirical learning with formal reasoning in autonomous artificial systems operating under partial observability.

Factual Acquisition Capacity. FAC evaluates an agent's ability to retrieve, validate, and integrate external knowledge from dynamic sources (e.g., web, files, APIs), building on prior work in factual grounding and knowledge retrieval accuracy. This capacity is fundamentally governed by the *tools* component, which include:

- **Tool Heterogeneity**: Diversity of integrated resources (e.g., search APIs, vision and audio modules) defining accessible knowledge domains.
- **Orchestration Scalability**: Architectural capacity to manage concurrent tool utilization and cross-modal data fusion.

Empirical boundaries emerge directly from toolset limitations, establishing hard constraints on factual knowledge acquisition.

Logical Reasoning Fidelity. LRF assesses an agent's capacity to maintain coherent causal reasoning and deduction chains during complex tasks, drawing from research on multi-step inference, reasoning robustness, and structured planning. The framework establishes formal foundations for sta-

ble and coherent decision-making through synergistic integration of three constitutive elements: *Plan*, *Memory*, and *Test-Time Scaling*. This triadic architecture manifests distinct operational principles per component:

- **Plan**: Maintains cognitive consistency through temporal synchronization between algorithmic planning strategies and memory-encoded experiential patterns.
- **Memory**: Ensures behavioral coherence through persistent state representations that anchor planning operations across decision episodes.
- **Test-Time Scaling**: Facilitates adaptive resilience by leveraging real-time performance diagnostics to dynamically recalibrate operational parameters.

3.1 Factual Acquisition Capacity (FAC)

Factual acquisition competence enables agents to systematically gather, verify, and integrate external knowledge via diverse tools. This capacity is fundamentally bounded by two critical operational vectors: multimodal tool interoperability and search tool efficacy, which jointly define the epistemic frontiers of agent-environment interactions.

We focus on quantifying current capability ceilings through two investigative lenses:

- *Multimodal tool constraints:* Characterizing temporal alignment errors and modality fusion bottlenecks in cross-domain information synthesis.
- *Search tool limitations:* Evaluating knowledge coverage gaps imposed by Search API constraints, index freshness thresholds, and semantic disambiguation failures in web-scale data retrieval.

3.1.1 Multimodal Toolkit

To address the limitations in contextual understanding faced by current agent systems, a multimodal toolkit is employed that integrates capabilities for processing text, speech, images, and video. Unlike traditional frameworks that rely solely on unimodal conversion to transform non-textual content into textual descriptions, this approach enables synchronized and cross-modal semantic parsing:

$$\text{Response} = \mathcal{A}(x_{\text{text}}, \mathcal{T}_{\text{image}}(I), \mathcal{T}_{\text{video}}(V)) \quad (1)$$

where $\mathcal{A}$ is the agent function, $x_{\text{text}}$is the textual input, and $\mathcal{T}_{\text{image}}, \mathcal{T}_{\text{video}}$ are tool functions that extract features from images I and videos V, respectively. This capability enhances the agent's ability to acquire and interpret factual information in complex, real-world scenarios through direct interaction with multimodal inputs.

3.1.2 Search Agent Framework

Web search enables LLM-agents to address real-time information needs and expand epistemic boundaries. We optimize three subsystems: *(i) Multi-source retrieval*, *(ii) Query refinement*, and *(iii) Minimalist browsing architecture* via the Search Agent framework.

Multi-Source Search. To mitigate single-source bias, we integrate commercial APIs (Google, Bing) and archival systems (Wayback Machine CDX API). Source selection is state-aware, driven by query temporal constraints (historical/real-time) and domain requirements (academic/commercial). Historical retrieval uses structured $\langle$url,date$\rangle$ queries to Internet Archive's temporal index.

Query Optimization Pipeline. Closed-loop refinement combines semantic calibration (REFLECT) with morphological expansion (EXPAND):

$$Q_{\text{opt}} = \text{REFLECT}(Q_{\text{init}}, M_{\text{task}}) \rightarrow \text{EXPAND}(Q_{\text{opt}}, L_{\text{term}}) \quad (2)$$

where REFLECT$(\cdot)$ resolves semantic ambiguities by calibrating specificity through prompt-based constraints and logical simplification guided by predefined rewrite rules, while EXPAND$(\cdot)$ generates morphological and semantic variants via stemming or lemmatization transformations, as well as domain-specific synonym expansion (e.g., *COVID-19* $\rightarrow$ *SARS-CoV-2*).

Minimalist Browsing. Conventional frameworks suffer from tool overload. We reduce complexity to three atomic functions: `Search (query)`: Find relevant web pages to the query from search engines. `Visit (url)`: Navigate to the webpage corresponding to url and `Read (url, mode)`: Extract contens in a page and present observations.

3.2 Logical Reasoning Fidelity (LRF)

In this section, we investigate three key strategies to improve logical reasoning in agents: dynamic plan generation and task decomposition, memory-augmented knowledge system, and test-time scaling for exploration optimization. These approaches address challenges in logical consistency, environmental adaptability, and efficiency-accuracy tradeoffs.

3.2.1 Dynamic Plan Generation

Strategic Plan Review. To enhance agents' complex task management, planning modules generate high-level plans $\mathcal{P} = (s_1, s_2, ..., s_n)$ that

decompose tasks into executable steps, improving reasoning efficiency. Execution follows the ReAct framework, alternating reasoning r_t and actions a_t. For adaptability in dynamic environments, plans are revised every N steps using recent observations $\{o_{t-N+1}, ..., o_t\}$): $\mathcal{P}' = \text{revise}(\mathcal{P}, \{o_{t-N+1}, ..., o_t\})$ This iterative planning-execution loop sustains goal-directed behavior and strengthens long-term decision-making.

Subtask Decomposition. To enhance systematic reasoning in planning modules, we propose hierarchical task decomposition: The agent breaks down the main goal $\mathcal{G}$ into interdependent subtasks $\mathcal{S} = (s_1, s_2, ..., s_n)$ and constructs a dependency graph $\mathcal{D} = (\mathcal{S}, \varepsilon)$, where edges $e_{ij} \in \varepsilon$ encode precedence constraints. At each reasoning step t, dynamic scheduling selects executable subsets $\mathcal{S}_t \subseteq \mathcal{S}$ satisfying all dependencies in $\mathcal{D}$. Intermediate outputs from completed subtasks are formalized as structured knowledge representations $_i \in \mathcal{K}$, which are cross-validated against global constraints $C(\mathcal{G})$. A validity function ensures alignment with the overarching goal:

$$\text{valid}(\kappa_i) = \begin{cases} \text{true}, & \text{if } \kappa_i \vDash C(G) \\ \text{false}, & \text{otherwise} \end{cases} \tag{3}$$

This mechanism enables error detection through consistency checks, strengthens long-horizon reasoning, and improves decision-making resilience in complex environments.

Plan Tips. To augment planning capabilities, we propose integrating experiential knowledge from historical execution trajectories $\tau\{(s_t, a_t, r_t)\}_t^T$. Analysis of past attempts reveals common bottlenecks and failure patterns, which are distilled into heuristic guidelines $\mathcal{H} = \{h_1, h_2, ..., h_m\}$ as soft constraints for the planner. These domain-specific heuristics influence action selection during planning through an augmented policy:

$$\pi_\theta(a_t \mid s_t, \mathcal{H}) = \text{softmax}(Q(s_t, a_t) + \beta \cdot f_{\mathcal{H}}(s_t, a_t)) \tag{4}$$

where $f_{\mathcal{H}}(\cdot)$ encodes the influence of heuristics and β controls their weight. This integration enables preemptive avoidance of known pitfalls, enhances robustness in plan generation, and improves adaptability to dynamic environments by embedding empirical knowledge into decision-making.

3.2.2 Memory-augmented Knowledge System

The hierarchical memory module enhances agent cognition through four components: *Current Memory*, *Memory Summarization*, *Vectorized Retrieval*, and *Long-Term Memory*, each addressing distinct aspects of perception and decision-making.

Current Memory. Serves as a short-term buffer storing temporally ordered task-specific information $M^c = \{(s_t, a_t)_{t-\tau}^t\}$, for real-time processing and on-the-fly decisions.

Memory Summarization. This component transforms raw experience sequences into structured semantic units using topic modeling and sequence-to-sequence generation:

$$z_i = \text{Summarize}(\{(s_t, a_t, r_t)_t^{t_{i+1}}\}) \tag{5}$$

where z_i denotes a memory summarization. By extracting high-salience knowledge, it facilitates efficient downstream processing.

Vectorized Memory Retrieval. This component retrieves beneficial historical memories via vector similarity. Specifically, the execution log of each step is embedded into a shared latent space $\mathcal{E}$: $\mathcal{E}(x) = \text{Encode}(x)$. Contextually relevant memories are then retrieved based on vector similarity:

$$M_{\text{retrieved}} = \arg\max_{m \in M} \text{sim}(\mathcal{E}(q), \mathcal{E}(m)) \tag{6}$$

Long-Term Memory. Addresses challenges in lengthy reasoning chains and contextual redundancy during task execution by integrating historical insights. Updates occur through fusion of current memory with existing long-term knowledge, enabling continuous optimization recommendations for task execution.

These components form a structured framework that organizes, stores, and retrieves knowledge at multiple abstraction levels, helping the agent perform effectively in complex environments.

3.2.3 Test-Time Scaling

The Test-Time Scaling (TTS) module enhances agent capabilities through three mechanisms: diversity enhancement, optimization, and reward modeling.

Diversity Enhancement. A mixture-of-agents sampling strategy combines multiple LLM policies π_{θ_i} with weights α_i:

$$a_t \sim \sum_{i=1}^{K} \alpha_i \cdot \pi_{\theta_i}(\cdot \mid s_t) \tag{7}$$

This exploits inter-model diversity to generate broader solution spaces and improve outcome quality.

Optimization. The TTS module guides agent reasoning through process-based reward functions $r_t = R(s_t, a_t)$ that assess task progression, error handling, and efficiency at each step. Rewards are temporally aggregated as:

$$R_{\text{total}} = \sum_{t=1}^{T} \gamma^t r_t \tag{8}$$

providing continuous feedback to refine reasoning trajectories and improve solution accuracy.

Reward Modeling. The TTS module enables real-time reflection for adaptive problem-solving through::

$$c_t = \text{Reflect}(\{(s_\tau, a_\tau)\}_{\tau=1}^{t}) \tag{9}$$

where c_t captures corrective insights from past steps, improving error detection and on-the-fly adjustments to enhance overall performance.

4 Empirical Study

4.1 Experimental Setup

GAIA(Mialon et al., 2023) presents real-world challenges that demand essential skills like reasoning, handling multi-modal inputs, web browsing, and overall proficiency in tool-calling. True answers are provided for each question, and the correctness of the model response is evaluated with exact match. Due to the instability and randomness of networked experiments, we allow the model to reanswer a question when the answer given by the model is empty or contains "Unable to determine" specified in the prompt. However, recalling incorrect answers is illegal. The evaluation protocol and implementation details can be found in Appendix A.

4.2 Main results

The results in Table 1 reveal several key insights into the performance landscape across various agent frameworks on the *GAIA* benchmark. Notably, our method (OAGENTS-Pass@3) achieves the highest overall average score of 73.93%, outperforming all other frameworks, including both closed-source and open-source systems. This highlights the robustness and effectiveness of our agent design.

In terms of Level 1 task performance, our method reaches 83.02%, tying with the best-performing frameworks and establishing a new standard for esay task handling. This superior performance reflects the reliability and consistency of our low-level agents and the underlying System Utilities.When compared with leading closed-source agents like Langfun Agent (71.52%) and TraseAgent (70.30%), our method shows a clear edge in both average and Level 2 accuracy. Finally, in the open-source domain, OAGENTS-Pass@3 demonstrates a significant margin over the best alternative, OWL-Roleplaying-Pass@3 (58.18%), reaffirming our method's leading position among publicly available systems. Overall, these results validate our approach as a state-of-the-art solution for generalist agent tasks.

We replicate Open Deep Research (LangChain, 2024) and note the results as "Smolagents", and the performance of the replication shows a significant degradation. This indicates that the reproducibility of the current agencies framework is poor.

4.2.1 FAC Evaluations

Multimodal Toolkit. We have refined text extraction tools with format-specific strategies tailored for various document types (*pdf*, *xlsx*, and *etc.*). For audio inputs, we employ the *whisper-1* speech-to-text model to generate accurate transcriptions. For video content, we implement a pipeline combining keyframe extraction with vision-language models for temporal and contextual analysis. Importantly, we incorporate a multi-source image understanding module, which leverages multiple vision language models source to understand visual features. Evaluated on the *GAIA* dataset (Table 2), our toolkit achieves a cross-modal task accuracy of 74.07%, outperforming the baseline system's 48.15%. Notably, in audio question-answering subtasks, temporal reasoning accuracy improves from 0% to 100% (3/3). These results demonstrate that a deeply optimized multimodal architecture can effectively bridge modality gaps in intelligent agent systems.

Findings 1

The multimodal toolkit's superiority stems from its ability to bridge modality-specific information gaps through synchronized semantic parsing.

Search Agent. Our empirical analysis quantitatively evaluates how search infrastructure design affects the performance of *GAIA*. As shown in Table 3, Jina reader outperforms raw HTML parsing by 9.3% in Level 2 tasks. Its structured text extraction benefits mid-complexity factual acquisition,

Table 2: Performance (%) of OAGENTS before and after integrating multimodal toolkit.

Method	GAIA multimodal tasks			
	Sum	Audio	Image	Tubular
Task number	27	3	10	14
OAGENTS	48.15	0.00	40.00	64.29
OAGENTS + Toolkit	74.07	100.00	60.00	78.57

highlighting preprocessing's role in enhancing retrieval quality.

Table 3: Performance comparison of browsing methods on *GAIA* benchmark. All results are obtained using information retrieved from Google Search.

Browsing Method	GAIA			
	Average	Level 1	Level 2	Level 3
Text web browser	44.20	54.71	43.02	26.92
Raw reader	49.70	64.51	46.51	30.76
Crawler crawl4ai	50.90	67.92	51.16	15.38
Jina reader	51.52	67.92	48.83	26.92

From Table 4, integrating complementary search engines (DuckDuckGo, Baidu, Bing) consistently improves retrieval accuracy, with the largest gain in Level 3 tasks (+7.69%). This indicates that diversifying information sources mitigates individual engine limitations, particularly in complex retrieval scenarios.

Table 4: OAGENTS performance of different search source configurations on *GAIA*. Note that "single-source" refers to Google only. "Multi-source (k = 3)" includes Google, Wikipedia, and DuckDuckGo. "multi-source (k = 5)" further adds Bing and Baidu as additional search sources.

Search Method	GAIA			
	Average	Level 1	Level 2	Level 3
Single-source	51.52	67.92	48.83	26.92
Multi-source (k = 3)	52.12	67.92	50.00	26.92
Multi-source (k = 5)	55.15	67.92	53.49	34.61

The proposed query optimization strategy, combining reflection and expansion mechanisms, significantly enhances system performance (Table 5). It yields a 7.55% improvement in Level 1 and 2.31% in Level 2, underscoring the effectiveness of refined query formulation in improving search outcomes. Finally, the minimalist system architecture demonstrates competitive performance, supporting the hypothesis that reduced interface complexity can improve robustness without sacrificing functionality.

Table 5: OAGENTS performance comparison of query-optimization configurations on *GAIA*.

Query Optimization	GAIA			
	Average	Level 1	Level 2	Level 3
Raw data	55.15	67.92	53.49	34.61
Reflection-Expansion	58.18	75.47	55.80	30.76

Findings 2

Multi-source retrieval and query optimization mitigate epistemic biases by diversifying knowledge acquisition and refining semantic precision.

OAGENTS. By integrating an optimized search infrastructure with a multimodal toolkit, and employing the Jina reader with multi-source (k = 5) strategies, our OAGENTS achieves strong improvement s on the *GAIA* benchmark across diverse base models. With GPT-4o, OAGENTS improves the overall score by 8.09%, including a 7.69% gain in Level 3 tasks. Gemini-2.5 shows a 9.09% average improvement, with Level 3 jumping 19.24%, confirming the effectiveness of the multimodal toolkit and refined search agent. Notably, Claude-3-7 gains 20.61%, the highest observed boost, demonstrating the framework's adaptability to models with varying baseline performance. The integrated design enhances FAC through advanced search and multimodal capabilities, establishing a solid foundation for knowledge-intensive agent systems. These results confirm that FAC improvements significantly elevate intelligent agent performance across architectures.

4.2.2 LRF Evaluations

Dynamic Plan Generation. The results in Table 7 show that our planning and workflow design significantly enhance GPT-4.1's ability to solve complex tasks. Strategic plan review (baseline) improves overall accuracy by 3.64% over the static workflow, confirming that dynamic plan revision supports better adaptability and long-term reasoning. Subtask Decomposition achieves a 2.42% improvement over baseline, demonstrating that breaking down tasks into structured subtasks enhances systematic reasoning, particularly for tasks of moderate complexity. The Plan tips are summarized from analysis of historical error logs and incorporate heuristic knowledge gained from past failures. They contribute to a 14.54% performance improvement, proving that leveraging prior expe-

Table 6: OAGENTS performance of various base models on *GAIA*.

Model	Type	GAIA Score			
		Average	Level 1	Level 2	Level 3
GPT-4o	Baseline	36.97	54.72	34.88	7.69
	Advance	45.06	62.26	45.35	15.38
	Gap	↑8.09	↑7.54	↑10.47	↑7.69
GPT-4.1	Baseline	44.20	54.71	43.02	26.92
	Advance	55.15	67.92	53.49	34.62
	Gap	↑10.95	↑13.21	↑10.47	↑7.70
OpenAI-o1	Baseline	49.70	54.72	53.49	26.92
	Advance	53.94	67.92	52.33	30.77
	Gap	↑4.24	↑13.20	↓1.16	↑3.85
Claude-3-7	Baseline	38.18	56.60	36.05	7.69
	Advance	58.79	64.15	61.63	38.46
	Gap	↑20.61	↑7.55	↑25.58	↑30.77
DeepSeek-R1	Baseline	33.90	45.28	33.72	11.54
	Advance	49.70	62.26	50.00	23.08
	Gap	↑15.80	↑16.98	↑16.28	↑11.54
Gemini-2.5	Baseline	49.09	69.81	46.51	15.38
	Advance	58.18	73.58	55.81	34.62
	Gap	↑9.09	↑3.77	↑9.30	↑19.24

rience helps prevent errors and build more robust plans. This is especially important for high complexity tasks. Together, these components significantly enhance the system's planning capabilities for complex reasoning.

Findings 3

Adaptive planning enhances long-horizon reasoning by balancing top-down task decomposition with bottom-up feedback integration.

Table 7: OAGENTS performance evaluation of plan studies on *GAIA*. Note that `Static workflow` refers to a scenario in which all tasks follow the same manually designed workflow.

Model combination	GAIA			
	Average	Level 1	Level 2	Level 3
OAGENTS	51.52	67.92	48.83	26.92
r.p. `Static workflow`	47.88	62.26	47.67	19.23
+ `Subtask`	53.94	71.70	51.16	26.92
+ `Plan tips`	66.06	79.25	66.28	38.46

Memory. From Figure 2, adding memory summarization slightly improved average accuracy from 51.52% to 52.12%. With memory retrieval, performance increased further to 53.33%. The most significant gain came from long-term memory, raising the average to 55.76%, while also achieving the competitive results across all difficulty levels. Memory summarization transforms raw experience sequences into distilled semantic units.

For Level 1 tasks, summarization is the most straightforward way to achieve optimal performance. However, the performance dip in Level 3 suggests that memory summarization erases too much details and the agent is unable to learn from past failures. Vectorized memory retrieval based on vector similarity enables the agent to retrieve contextually relevant memories efficiently. The performance of vector retrieval in OAGENTS may be limited by the memory slicing approach and the prior knowledge of the encoder model. Long-term memory integrates historical insights and generates optimization recommendations for task execution. This allows OAGENTS to continuously learn and improve over time, based on its accumulated experiences. By fusing current memory with existing long-term memory, OAGENTS can leverage both recent and past knowledge to make more informed decisions, which is crucial for complex reasoning tasks that require a deep understanding of the context and history.

Findings 4

Hierarchical memory enhances context retention by separating short-term processing from long-term knowledge preservation.

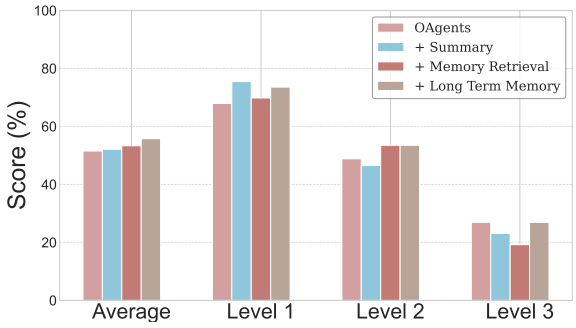

Figure 2: OAGENTS performance evaluation of various memory methods on *GAIA*.

Test-Time Scaling. As shown in Figure 3, we conduct an ablation study to examine how test-time scaling (TTS) strategies influence the performance of OAGENTS across different task complexities. Reflection leads to a moderate overall improvement (3.03%), yet its effects vary across task levels. While it enhances performance on Level 1 and Level 2 tasks through iterative reasoning, it unexpectedly degrades results on Level 3 tasks by 6.62%, suggesting potential instability or error accumulation in complex reasoning chains.

Best-of-N sampling demonstrates more consistent gains, with performance improving as the sample size increases. BO2 yields modest improvements (1.82%), while BO4 achieves the best overall performance (5.19%), particularly benefiting simpler tasks (Level 1: 9.44%, Level 2: 10.46%). This indicates that answer diversification helps in navigating simpler solution spaces more effectively.

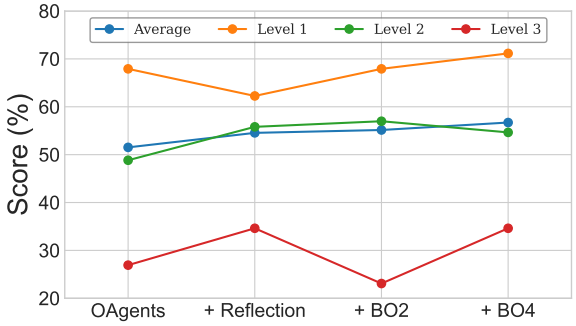

Figure 3: OAGENTS performance evaluation of TTS methods on *GAIA*.

Nonetheless, neither strategy substantially improves performance on Level 3 tasks, underscoring the persistent difficulty in achieving robust multi-step reasoning at scale. These findings reveal that TTS strategies exhibit differential effectiveness depending on task complexity—offering clear benefits for straightforward tasks but requiring further innovation to address advanced reasoning challenges.

Findings 5

Test-time scaling improves solution robustness by leveraging model diversity and iterative self-refinement.

5 *GAIA* Benchmark

The *GAIA* benchmark has emerged as a prominent evaluation framework for assessing the performance of autonomous agents in real-world scenarios. As the leaderboard for this benchmark continues to grow, it becomes increasingly evident that reported results often vary in terms of evaluation metrics—particularly in the use of different *Pass@K* criteria. While some methods report *Pass@1*, others adopt more lenient metrics such as *Pass@3* or even *Pass@5*. This inconsistency complicates fair comparisons across different agent frameworks and limits the transparency of their actual capabilities.

To address this issue and ensure alignment with the leaderboard standards, we reimplement the OWL-Roleplaying framework to obtain its *Pass@1* performance for comparison. Additionally, we evaluated our proposed open-source framework, OAGENTS, under the *Pass@3* setting, as summarized in Table 8. Built upon integrated multi-modal toolkit, multi-source information retrieval, and test-time scaling (TTS) strategies, OAGENTS demonstrates competitive performance among existing open-source frameworks under the *Pass@3* metric. These results highlight the framework's effectiveness in handling complex reasoning tasks and its strong potential for deployment in real-world applications requiring robust and scalable reasoning capabilities.

Table 8: Comparison of performance on the *GAIA* benchmark under different *Pass@K* metrics. Note that "OWL" stands for the open-source role-playing version.

Method	Model	Metric	GAIA			
			Average	Level 1	Level 2	Level 3
OAGENTS	Claude-3-7	Pass@1	**66.67**	**77.36**	**66.28**	**46.15**
OWL	4o & o3-mini		53.33	71.70	50.00	26.92
AWorld	Claude-3-7		61.81	-	-	-
OAGENTS	Claude-3-7	Pass@3	**73.93**	**83.02**	**74.42**	**53.85**
OWL	4o & o3-mini		58.18	81.14	54.65	23.08
AWorld	Claude-3-7	Unknown	77.58	88.68	77.91	53.85

6 Conclusion

In this work, we conduct a systematic study on the *GAIA* benchmark. We identify key components for effective agents, such as planning, memory, and tool use, and propose a robust evaluation protocol. We release OAGENTS, an open-source modular agent framework achieves state-of-the-art performance on *GAIA* (**73.93**), providing a foundation for future research on agent systems.

Limitations

The analysis reveals three core limitations in evaluating reasoning capabilities. Persistent outcome instability across models and task complexities emerges, particularly undermining multi-hop reasoning robustness. While multi-source retrieval enhances factual grounding, infrastructural dependencies constrain knowledge fidelity due to inherent search architecture limitations. Moreover, benchmark-centric evaluations (e.g., *GAIA*) may lack ecological validity, as structured test environments inadequately capture the non-linear cognitive processes required for authentic problem-solving scenarios, necessitating adaptive frameworks for dynamic real-world cognition assessment.

References

Agent Team at Ant Group. 2025. Aworld: A unified agent playground for computer and phone use tasks.

Dzmitry Bahdanau, Nicolas Gontier, Gabriel Huang, Ehsan Kamalloo, Rafael Pardinas, Alex Piché, Torsten Scholak, Oleh Shliazhko, Jordan Prince Tremblay, Karam Ghanem, Soham Parikh, Mitul Tiwari, and Quaizar Vohra. 2024. Tapeagents: a holistic framework for agent development and optimization. *Preprint*, arXiv:2412.08445.

Adam Fourney, Gagan Bansal, Hussein Mozannar, Cheng Tan, Eduardo Salinas, Friederike Niedtner, Grace Proebsting, Griffin Bassman, Jack Gerrits, Jacob Alber, and 1 others. 2024. Magentic-one: A generalist multi-agent system for solving complex tasks. *arXiv preprint arXiv:2411.04468*.

H2O.ai. 2024. Autonomous agentic ai: execute multi-step workflows autonomously. [Online]. https://h2o.ai/platform/enterprise-h2ogpte/#AgenticAI.

Mengkang Hu, Yuhang Zhou, Wendong Fan, Yuzhou Nie, Bowei Xia, Tao Sun, Ziyu Ye, Zhaoxuan Jin, Yingru Li, Zeyu Zhang, Yifeng Wang, Qianshuo Ye, Ping Luo, and Guohao Li. 2025. Owl: Optimized workforce learning for general multi-agent assistance in real-world task automation.

LangChain. 2024. Open deep research. [Online]. https://github.com/langchain-ai/open_deep_research.

Guohao Li, Hasan Hammoud, Hani Itani, Dmitrii Khizbullin, and Bernard Ghanem. 2023. Camel: Communicative agents for" mind" exploration of large language model society. *Advances in Neural Information Processing Systems*, 36:51991–52008.

Na Liu, Liangyu Chen, Xiaoyu Tian, Wei Zou, Kaijiang Chen, and Ming Cui. 2024. From llm to conversational agent: A memory enhanced architecture with fine-tuning of large language models. *Preprint*, arXiv:2401.02777.

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

Long Phan, Alice Gatti, Ziwen Han, Nathaniel Li, Josephina Hu, Hugh Zhang, Chen Bo Calvin Zhang, Mohamed Shaaban, John Ling, Sean Shi, and 1 others. 2025. Humanity's last exam. *arXiv preprint arXiv:2501.14249*.

Yujia Qin, Shengding Hu, Yankai Lin, Weize Chen, Ning Ding, Ganqu Cui, Zheni Zeng, Xuanhe Zhou, Yufei Huang, Chaojun Xiao, and 1 others. 2024. Tool learning with foundation models. *ACM Computing Surveys*, 57(4):1–40.

Aymeric Roucher, Albert Villanova del Moral, Thomas Wolf, Leandro von Werra, and Erik Kaunismäki. 2025. 'smolagents': a smol library to build great agentic systems. https://github.com/huggingface/smolagents.

Noah Shinn, Federico Cassano, Edward Berman, Ashwin Gopinath, Karthik Narasimhan, and Shunyu Yao. 2023. Reflexion: Language agents with verbal reinforcement learning. *Preprint*, arXiv:2303.11366.

Significant-Gravitas. 2023. Autogpt. [Online]. https://github.com/Significant-Gravitas/AutoGPT.

Chan Hee Song, Jiaman Wu, Clayton Washington, Brian M Sadler, Wei-Lun Chao, and Yu Su. 2023. Llm-planner: Few-shot grounded planning for embodied agents with large language models. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 2998–3009.

Jiabin Tang, Tianyu Fan, and Chao Huang. 2025. Autoagent: A fully-automated and zero-code framework for llm agents. *arXiv e-prints*, pages arXiv–2502.

Jesus Tordesillas and Jonathan P How. 2021. Mader: Trajectory planner in multiagent and dynamic environments. *IEEE Transactions on Robotics*, 38(1):463–476.

Trase. 2024. Meet trase systems. [Online]. https://www.trasesystems.com/.

Xingyao Wang, Yangyi Chen, Lifan Yuan, Yizhe Zhang, Yunzhu Li, Hao Peng, and Heng Ji. 2024. Executable code actions elicit better llm agents. In *Forty-first International Conference on Machine Learning*.

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

Qingyun Wu, Gagan Bansal, Jieyu Zhang, Yiran Wu, Beibin Li, Erkang Zhu, Li Jiang, Xiaoyun Zhang, Shaokun Zhang, Jiale Liu, and 1 others. 2023. Autogen: Enabling next-gen llm applications via multi-agent conversation. *arXiv preprint arXiv:2308.08155*.

Tianbao Xie, Fan Zhou, Zhoujun Cheng, Peng Shi, Luoxuan Weng, Yitao Liu, Toh Jing Hua, Junning Zhao, Qian Liu, Che Liu, and 1 others. 2023. Openagents: An open platform for language agents in the wild. *arXiv preprint arXiv:2310.10634*.

Wujiang Xu, Zujie Liang, Kai Mei, Hang Gao, Juntao Tan, and Yongfeng Zhang. 2025. A-mem: Agentic memory for llm agents. *arXiv preprint arXiv:2502.12110*.

Dingkang Yang, Kun Yang, Yuzheng Wang, Jing Liu, Zhi Xu, Rongbin Yin, Peng Zhai, and Lihua Zhang. 2023a. How2comm: Communication-efficient and

collaboration-pragmatic multi-agent perception. *Advances in Neural Information Processing Systems*, 36:25151–25164.

Kun Yang, Dingkang Yang, Jingyu Zhang, Hanqi Wang, Peng Sun, and Liang Song. 2023b. What2comm: Towards communication-efficient collaborative perception via feature decoupling. In *Proceedings of the 31st ACM international conference on multimedia*, pages 7686–7695.

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

Zeyu Zhang, Xiaohe Bo, Chen Ma, Rui Li, Xu Chen, Quanyu Dai, Jieming Zhu, Zhenhua Dong, and Ji-Rong Wen. 2024. A survey on the memory mechanism of large language model based agents. *arXiv preprint arXiv:2404.13501*.

Wangchunshu Zhou, Yuchen Eleanor Jiang, Peng Cui, Tiannan Wang, Zhenxin Xiao, Yifan Hou, Ryan Cotterell, and Mrinmaya Sachan. 2023a. Recurrentgpt: Interactive generation of (arbitrarily) long text. *arXiv preprint arXiv:2305.13304*.

Wangchunshu Zhou, Yuchen Eleanor Jiang, Long Li, Jialong Wu, Tiannan Wang, Shi Qiu, Jintian Zhang, Jing Chen, Ruipu Wu, Shuai Wang, Shiding Zhu, Jiyu Chen, Wentao Zhang, Xiangru Tang, Ningyu Zhang, Huajun Chen, Peng Cui, and Mrinmaya Sachan. 2023b. Agents: An open-source framework for autonomous language agents.

Wangchunshu Zhou, Yixin Ou, Shengwei Ding, Long Li, Jialong Wu, Tiannan Wang, Jiamin Chen, Shuai Wang, Xiaohua Xu, Ningyu Zhang, Huajun Chen, and Yuchen Eleanor Jiang. 2024. Symbolic learning enables self-evolving agents.

A Experimental Details

Evaluation Protocol. We follow the evaluation protocol of the *GAIA*benchmark (Mialon et al., 2023), which is based on exact match accuracy. The primary metric used is *Pass@N*, which measures the probability that at least one correct solution is found among N independent model attempts. This metric is widely adopted in tasks such as code generation, where the key evaluation criterion is whether the model can produce a valid solution at least once. In our experiments, unless otherwise stated, we report the average *Pass@1* score, reflecting the model's performance in generating a correct answer across all questions within a single evaluation run.

Implementation Details. In both the FAC Evaluations and LRC Evaluations, the baselines are implemented with the integrated multimodal toolkit in OAGENTS. Unless otherwise specified, all models employed in the agent are based on GPT-4.1 to ensure consistency in model architecture and capabilities across experiments.

B Additional Evaluations

B.0.1 Generalizability Evaluation

To investigate the generalizability, we evaluate OAgents using two challenging datasets named BrowserComp (Wei et al., 2025) and HLE (Phan et al., 2025), on which the base model rarely answered correctly or scored. As shown in Table 9, our OAgents significantly improved the model's abilities in search and information integration.

Table 9: The performance of OAGENTS on BrowseComp-Subset and HLE-Subset.

Model	BrowserComp-Subset	HLE-Subset
Claude-3-7	4.76%	8.00%
GPT-4.1	7.94%	5.40%
OpenAI-o1	14.29%	8.00%
OAGENTS - GPT-4.1	**22.22%**	**15.43%**
OAGENTS - Claude-3-7	**22.22%**	**14.86%**

B.1 Statistical Validation

In Table 10, we report the performance of three variants of OAGENTS at avg@3 and their performance in each run. Where "OAgents w/ Claude3-7" corresponds to Table 1, "Advance w/ GPT-4.1" corresponds to Table 4, 5, and 6. "LRF Baseline" is the baseline method for all LRF Evaluations including Table 4, 7, and 8, Figure 2 and 3.

C Details of OAGENTS

C.1 Search Agent

Web search constitutes a foundational capability for LLM-agents to address real-time information needs and extend their epistemic boundaries. We focus on optimizing three critical subsystems: *(i) Multi-source retrieval*, *(ii) Query refinement*, and *(iii) Adaptive browsing* – implemented through the SearchAgent framework.

Multi-Source Search. Contemporary search engines exhibit non-overlapping ranking mechanisms and temporal coverage limitations. To mitigate single-source bias, our implementation integrates:

- Commercial APIs: Google Custom Search JSON API and Bing Web Search API.
- Archival Systems: Wayback Machine CDX Server API for historical snapshots.

In a state-aware routing mechanism, source selection is autonomously driven by:

- Query temporal constraints (historical vs. real-time).
- Domain-specific coverage requirements (academic vs. commercial).

Table 10: Statistical evaluation of OAGENTS

Agents	Run	Overall	Level 1	Level 2	Level 3
OAgents w/ Claude3-7	Run 1	66.67	77.36	66.28	46.15
	Run 2	63.64	77.36	61.63	42.31
	Run 3	62.42	77.36	62.79	30.77
	Avg@3	64.24 ± 2.19	77.36 ± 0.00	63.57 ± 2.42	39.74 ± 8.00
Advance w/ GPT-4.1	Run 1	55.15	67.92	53.49	34.62
	Run 2	56.97	69.81	55.81	34.62
	Run 3	56.97	69.81	54.65	38.46
	Avg@3	56.36 ± 1.05	69.18 ± 1.09	54.65 ± 1.16	35.90 ± 2.22
LRF Baseline	Run 1	51.52	67.92	48.83	26.92
	Run 2	48.48	60.38	47.67	26.92
	Run 3	50.30	62.26	47.67	34.62
	Avg@3	50.10 ± 1.53	63.52 ± 3.92	48.07 ± 0.67	29.49 ± 4.44

The historical retrieval tool accepts structured inputs as $<url, date>$ tuples, querying the Internet Archive's temporal index through:

Listing 1: Example of construct a CDX query to retrieval archive information.

```
def fetch_historical_page(url: str, timestamp: str) -> str:
    cdx_query = f"http://web.archive.org/cdx/search/cdx?url={url}&output=json&from={
        timestamp}"
```

Browsing Method. In Table 3, we introduce four browsing methods, including Text web browser, Raw reader, Crawler, and Jina reader, among which Jina reader performs the best. We would like to make further explanations here:

- Text web browser is implemented by Smolagents, which feeds paged html files to the model and lets the model navigate through the web by paging up and down. This approach is not suitable for knowledge-intensive tasks because it consumes a large number of requests and challenges the model's memory and planning capabilities.
- In Raw reader, we enter all the searched web pages into the model at once. The web pages are still in HTML format, the model is still responsible for parsing the pages implicitly, and the HTML code blocks consume a sizable context window.
- We build Crawler using crawl4ai to extract rich text from HTML and simplify the model's optional operations to query, visit, and read. However, we found that crawl4ai is unable to access or parse some domain-specific web pages, such as Arxiv and Youtube.
- Jina has stronger access and parsing ability compared to crawl4ai, we construct Jina reader in a similar way to Crawler, it works best among all browsing methods.

Query Optimization Pipeline. The closed-loop query refinement follows:

$$Q_{\text{opt}} = \textsc{Reflect}(Q_{\text{init}}, M_{\text{task}}) \rightarrow \textsc{Expand}(Q_{\text{opt}}, L_{\text{term}}) \tag{10}$$

where $\textsc{Reflect}(\cdot)$ resolves semantic ambiguities by calibrating specificity through prompt-based constraints and logical simplification guided by predefined rewrite rules, while $\textsc{Expand}(\cdot)$ generates morphological and semantic variants via stemming or lemmatization transformations, as well as domain-specific synonym expansion (e.g., *COVID-19* → *SARS-CoV-2*).

Minimalist browsing architecture. Conventional browser emulation frameworks impose cognitive overhead through excessive tool options. Our streamlined implementation reduces interaction complexity by:

- Eliminate non-essential operations (e.g., click, scroll, find).
- Consolidate functionality into three atomic tools: `Search`, `Visit`, and `Read`.

C.2 Strategic Plan Review

In order to improve an agent's capability to manage complex tasks, the incorporation of a planning module is of critical importance. Planning module enables the agent to generate a high-level plan $\mathcal{P} = (s_1, s_2, ..., s_n)$ before execution, breaking down complex tasks into manageable steps and improving reasoning efficiency. Execution typically follows the ReAct framework, interleaving reasoning and actions: at each step t, the agent performs either an action a_t or a reasoning step r_t. To ensure adaptability in dynamic environments, the plan $\mathcal{P}$ is periodically revised—every N steps—based on new observations o_t, updating the sequence of subtasks as $\mathcal{P}' = \text{revise}(\mathcal{P}, \{o_{t-N+1}, ..., o_t\})$ This iterative approach supports sustained, goal-directed behavior and enhances the agent's long-term reasoning and decision-making capabilities.

C.3 Subtask Decomposition.

Given the role of the planning module in managing complex tasks, we can further consider a hierarchical task decomposition mechanism to enhance systematic reasoning. During planning, the agent decomposes the main goal $\mathcal{G}$ into a set of interdependent subtasks $\mathcal{S} = (s_1, s_2, ..., s_n)$, and constructs a dependency graph $\mathcal{D} = (\mathcal{S}, \varepsilon)$, where edges $e_{ij} \in \varepsilon$ represent precedence constraints between subtasks. This structure enables dynamic scheduling of non-conflicting subtasks at each reasoning step t, formalized as selecting an executable subset $\mathcal{S}_t \subseteq \mathcal{S}$ such that all dependencies in $\mathcal{D}$ are satisfied. A key component is the iterative synthesis of intermediate outputs: results from completed subtasks are formalized as structured knowledge representations $_i \in \mathcal{K}$, and refined through cross-validation against global constraints $C(\mathcal{G})$. This process ensures alignment with the overarching goal and supports error detection and correction via consistency checks:

$$\text{valid}(\kappa_i) = \begin{cases} \text{true}, & \text{if } \kappa_i \vDash C(G) \\ \text{false}, & \text{otherwise} \end{cases} \tag{11}$$

Collectively, these mechanisms strengthen the planning module's capacity for long-horizon reasoning, enabling more effective and resilient decision-making in complex environments.

C.4 Plan Tips.

Beyond designing diverse planning strategies, another promising direction lies in enriching the planning process with additional prior knowledge. By analyzing the execution trajectories $\tau\{(s_t, a_t, r_t)\}_t^T$ of past attempts, w, we can identify common bottlenecks, failure points, and suboptimal behaviors encountered by the agent during task realization. These insights can then be distilled into actionable tips or heuristic guidelines $\mathcal{H} = \{h_1, h_2, ..., h_m\}$, which are subsequently injected into the planning module as soft constraints or preferences.

Such domain-specific knowledge serves as supplementary guidance during plan generation, influencing the selection of actions and subgoals:

$$\pi_\theta(a_t \mid s_t, \mathcal{H}) = \text{softmax}(Q(s_t, a_t) + \beta \cdot f_{\mathcal{H}}(s_t, a_t)) \tag{12}$$

where $f_{\mathcal{H}}(\cdot)$ encodes the influence of heuristics and β controls their weight. As a result, the planner is better equipped to anticipate potential issues, avoid known pitfalls, and construct more robust strategies for complex problem-solving. This integration of experiential knowledge enhances not only the effectiveness of individual planning steps but also the overall resilience of the agent in dynamic and uncertain environments.

C.5 Memory-augmented Knowledge System

The hierarchical memory module is designed to enhance the cognitive capabilities of intelligent agents through four complementary components: *Current memory*, *Memory summarization*, *Memory retrieval*, and *Long-term memory*. Each component contributes uniquely to different aspects of perception, reasoning, and decision-making.

C.5.1 Current Memory.

As a fundamental default component of the agent, current memory acts as a short-term buffer to capture fine-grained, task-specific information in real time. This buffer maintains recent observations and actions in a temporal sequence $M^c = \{(s_t, a_t)_{t-\tau}^{t}\}$, enabling the agent to process dynamic environmental inputs with high fidelity and support on-the-fly decision-making.

C.5.2 Memory Summarization.

This component transforms raw experience sequences into structured semantic units using topic modeling and sequence-to-sequence generation:

$$z_i = \text{Summarize}(\{(s_t, a_t, r_t)_t^{t_{i+1}}\}) \tag{13}$$

where z_i denotes a memory summarization. By extracting high-salience knowledge, it facilitates efficient downstream processing.

C.5.3 Vectorized Memory Retrieval.

This component retrieves beneficial historical memories via vector similarity. Specifically, the execution log of each step is embedded into a shared latent space $\mathcal{E}$: $\mathcal{E}(x) = \text{Encode}(x)$. Contextually relevant memories are then retrieved based on vector similarity:

$$M_{\text{retrieved}} = \arg\max_{m \in M} \text{sim}(\mathcal{E}(q), \mathcal{E}(m)) \tag{14}$$

C.5.4 Long-Term Memory.

This component is designed to address the challenges of lengthy reasoning chains and redundant contextual information when agents perform tasks by integrating key insights from historical reasoning processes and generating subsequent optimization recommendations. Specifically, the long-term memory component achieves updates by fusing current memory with existing long-term memory, continuously guiding agents in task execution.

C.6 Test-Time Scaling

Agent capabilities can be significantly enhanced through the integration of test-time scaling mechanisms, which dynamically refine decision-making, improve adaptability, and promote more robust exploration. Test-Time-Scaling (TTS) module contributes to this enhancement by addressing three core aspects: diversity, optimization, and reward modeling.

C.6.1 Diversity Enhancement.

Enhancing the diversity of reasoning paths is crucial for improving agent performance in complex tasks. By leveraging a mixture-of-agents sampling strategy:

$$a_t \sim \sum_{i=1}^{K} \alpha_i \cdot \pi_{\theta_i}(\cdot \mid s_t) \tag{15}$$

where α_i denotes the weight of each agent policy π_{θ_i}, the TTS module exploits differences in capability profiles across multiple LLMs, generating a broader range of potential solutions and increasing the likelihood of identifying high-quality outcomes.

C.6.2 Optimization.

To guide agents toward more effective reasoning trajectories, the TTS module introduces process based reward functions $r_t = R(s_t, a_t)$, which evaluate each step along the generation path. These multi dimensional assessments cover key aspects such as task progression, error handling, and efficiency. The rewards are aggregated over time:

$$R_{\text{total}} = \sum_{t=1}^{T} \gamma^t r_t \tag{16}$$

providing fine-grained feedback that enables iterative refinement and convergence toward more accurate final responses.

C.6.3 Reward Modeling.

Real-time reflection and self-correction are essential for adaptive problem-solving. The TTS module incorporates a reflection mechanism that evaluates intermediate steps during exploration:

$$c_t = \text{Reflect}(\{(s_\tau, a_\tau)\}_{\tau=1}^{t}) \tag{17}$$

where c_t represents corrective insights fed back into subsequent reasoning stages. This iterative refinement enhances the agent's ability to detect and rectify errors on-the-fly, leading to improved overall performance.