YUNQUE DEEPRESEARCH TECHNICAL REPORT

Yuxuan Cai[1], **Xinyi Lai**[1], **Peng Yuan**[1,2], **Weiting Liu**[1,3], **Huajian Li**[1], **Mingda Li**[1], **Xinghua Wang**[1], **Shengxie Zheng**[1], **Yanchao Hao**[1], **Yuyang Yin**[1*], **Zheng Wei**[1*]

[1]Tencent BAC [2]Tsinghua University [3]Fudan University

ABSTRACT

Deep research has emerged as a transformative capability for autonomous agents, empowering Large Language Models to navigate complex, open-ended tasks. However, realizing its full potential is hindered by critical limitations, including escalating contextual noise in long-horizon tasks, fragility leading to cascading errors, and a lack of modular extensibility. To address these challenges, we introduce **Yunque DeepResearch**, a hierarchical, modular, and robust framework. The architecture is characterized by three key components: (1) a centralized *Multi-Agent Orchestration System* that routes subtasks to an *Atomic Capability Pool* of tools and specialized sub-agents; (2) a *Dynamic Context Management* mechanism that structures completed sub-goals into semantic summaries to mitigate information overload; and (3) a proactive *Supervisor Module* that ensures resilience through active anomaly detection and context pruning. Yunque DeepResearch achieves state-of-the-art performance across a range of agentic deep research benchmarks, including GAIA, BrowseComp, BrowseComp-ZH, and Humanity's Last Exam. We open-source the framework, reproducible implementations, and application cases to empower the community.

1 Introduction

The pursuit of Artificial General Intelligence (AGI) has traditionally prioritized scaling Large Language Models (LLMs) to internalize vast repositories of passive knowledge. However, a critical limitation remains: while these models demonstrate exceptional proficiency in information recall and text generation, they lack the intrinsic agency required to actively discover, verify, and synthesize insights from dynamic, real-world environments. This constraint has necessitated a paradigm shift from static conversational models to autonomous agentic systems capable of grounded problem-solving[1, 2, 3]. Central to this evolution is Deep Research—a capability that empowers agents to orchestrate multi-step reasoning, utilize external tools such as web browsers, and navigate open-ended information landscapes autonomously[4, 5, 6, 7, 8]. By bridging the gap between pre-trained parameters and active inquiry, Deep Research agents represent a fundamental advancement toward systems that not only emulate human research workflows but substantially augment intellectual productivity.

Despite this promise, realizing the full potential of Deep Research presents significant challenges that prevalent monolithic agent architectures fail to address adequately: **(1) Cognitive Overload in Long-Horizon Tasks** — Long-horizon tasks require maintaining contextual continuity over hundreds of interaction steps. However, in prevailing ReAct-based agents[1, 9], the accumulation of raw execution logs often dilutes the original user intent and degrades reasoning performance. **(2) Systemic Fragility and Cascading Failures** — Existing systems often lack resilience. In the absence of robust error detection and recovery mechanisms, minor errors can trigger cascading failures, frequently trapping the entire system in recursive, suboptimal loops[10, 11]. **(3) Lack of Modular Extensibility** — The rigid architecture of many contemporary agents inhibits the flexibility required for diverse real-world applications. As research tasks grow in complexity, the inability to seamlessly integrate specialized tools or domain-specific sub-agents limits system composability and adaptability in evolving information environments[12].

To address these challenges, we introduce **Yunque DeepResearch**, a hierarchical agentic framework explicitly engineered to navigate the complexities of deep research. Our architecture fundamentally restructures the research

Code Repository: `https://github.com/Tencent-BAC/YunqueAgent` workflow through a decoupled design that fosters inherent modularity and extensibility, integrating effective sub-agent orchestration and long-horizon context management to transform the typically fragile linear chain into a robust and stable operation. Our system is built upon the following core design principles:

- **Effective Orchestration System.** We implement a centralized orchestration framework anchored by a Main Agent that serves as the strategic core. Utilizing a flexible dispatch mechanism, the planner dynamically routes tasks to the most appropriate resource within the Atomic Capability Pool: it directly invokes basic tools for low-latency atomic operations while delegating complex, long-horizon objectives to specialized sub-agents.
- **Dynamic Context Management.** We propose a sub-goal-driven memory mechanism to resolve the tension between context length and information density. By treating sub-goals as the fundamental unit of trajectory segmentation, our system dynamically partitions the research process: completed sub-goals are folded into concise structured summaries to maintain global planning awareness, while the active sub-goal retains fine-grained ReAct traces for precise execution. This hybrid approach transforms linear history into structured semantic milestones.
- **Modularity and Extensibility.** We ensure adaptability through a modular "Atomic Capability Pool" that separates strategic planning from action execution. By standardizing basic tools and specialized sub-agents as functional units, our architecture attains high composability. This separation creates an extensible ecosystem where new capabilities—ranging from atomic utility functions to expert-level solvers—can be dynamically registered and deployed, ensuring the framework remains resilient to evolving requirements.
- **Stability and Robustness.** We incorporate a dedicated Supervisor module to ensure system stability and mitigate the fragility often seen in long-horizon tasks. Unlike rigid reflection schedules, this mechanism performs active anomaly detection on the agent's trajectory. Upon identifying failures, it triggers a self-correction protocol, explicitly prunes invalid context to prevent memory pollution, guiding the agent to autonomously recover and synthesize a viable alternative response.

2 Related Work

DeepResearch Agent Recent advancements have witnessed the rapid emergence of Deep Research Agents—notably OpenAI's Deep Research[13] and Gemini Deep Research[14]—which synergize dynamic reasoning, adaptive planning, and multi-round evidence retrieval to tackle sophisticated open-domain challenges. Within the current research landscape, these systems are objectively categorized into single-agent and multi-agent paradigms based on their architectural orchestration. Conventional single-agent models [13, 14, 8] typically employ monolithic control architectures to manage tool orchestration. However, such centralized decision-making often suffers from "reliability bottlenecks" and cognitive saturation, particularly when navigating expansive context windows or filtering high-entropy web data. This centralization limits the agent's ability to maintain long-horizon reasoning consistency, leading to performance degradation in complex, multi-step research trajectories. Conversely, multi-agent frameworks [15, 16, 17, 18, 19] address these limitations by decomposing monolithic objectives into granular, manageable sub-tasks. By leveraging a collective of autonomous, heterogeneous agents—such as dedicated planners, retrievers, and critics—these systems facilitate modular collaboration and explicit error-correction loops. This transition from individual execution to collective orchestration enables superior handling of complex dependencies, effectively mitigating the error propagation typically observed in single-agent configurations.

Working Memory Management Effective management of working memory is crucial to prevent agents from falling into strategic redundancy or infinite loops. Early approaches, such as ReSum[20], utilize specialized summarization models to distill history upon reaching context limits. Subsequent works, including MeM1[21] and Memory-as-Action[22], treat memory management as an explicit reasoning action, enabling agents to autonomously summarize or edit their internal states. To handle ultra-long sequences, MemAgent[23] selectively stores task-relevant information, while AgentFold[24] adaptively determines compression granules to balance information density. Despite these advances, existing methods primarily focus on content compression rather than structured synthesis, often failing to maximize the effective information density of the agent's historical trajectory.

Specialized Agent Beyond general-purpose research, specialized agents have been developed for domain-specific tasks. GUI Agents (e.g., CogAgent[25], Claude 3.5 Sonnet[26]) autonomously interact with digital systems by perceiving and manipulating visual elements. Similarly, Data Analysis Agents (e.g., OpenAI's Advanced Data Analysis[27], Data Interpreter[28]) automate the end-to-end pipeline of data cleaning, processing, and reporting. Notably, these specialized agents are increasingly integrated as functional modules within larger multi-agent frameworks (e.g., OpenAgents[29], AgentOrchestra[16]) to execute sophisticated data-centric sub-goals.

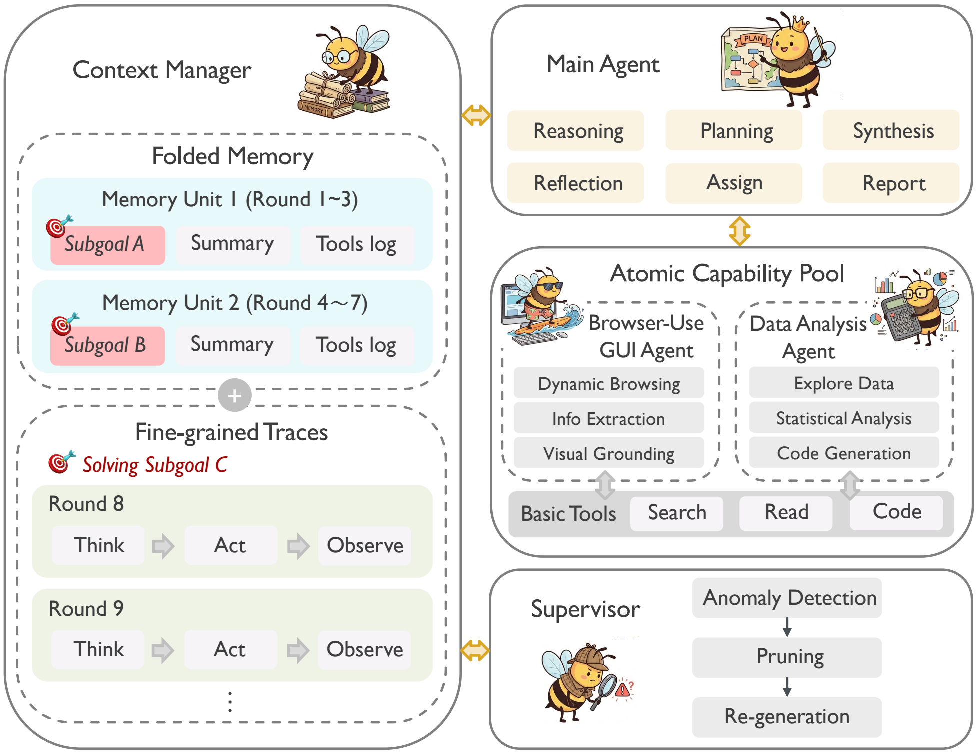


Figure 1: Overview of Yunque DeepResearch.

3 Framework

As illustrated in Figure 1, we propose Yunque DeepResearch, a hierarchical architecture designed to explicitly decouple atomic capabilities from the central reasoning process. The framework comprises four collaborative modules: *(i) Main Agent, (ii) Context Manager, (iii) Atomic Capability Pool, (iv) Supervisor*.

- **Main Agent**: At the core of the system, this module acts as the central executive, responsible for high-level intent recognition, dynamic planning, global orchestration, and result synthesis.
- **Context Manager**: To support navigating long-horizon tasks, this module establishes a dual-level memory structure that balances immediate operational precision with long-term strategic context.
- **Atomic Capability Pool**: Complementing the reasoning core, this pool hosts a diverse set of specialized sub-agents (e.g., Browser-Use GUI and Data Analysis agents) and fundamental tools, enabling the execution of specific atomic actions.
- **Supervisor**: This module ensures system robustness by monitoring the execution trajectory, performing error correction, and preventing cascading failures during multi-turn interactions.

3.1 Main Agent

The Main Agent focuses on decomposing high-level user instructions into manageable sub-tasks through intent analysis and logical structuring. Adopting an interleaved reasoning paradigm, the agent actively interprets feedback from each step to dynamically refine its subsequent plans, persisting reasoning context across tool invocations and sub-agent handoffs to maintain global coherence.

To execute these plans, the Main Agent employs an adaptive routing mechanism that synergizes lightweight tools with specialized sub-agents. Instead of adhering to a rigid division of labor, the agent dynamically orchestrates these complementary resources based on the evolving context. This design allows for a fluid interplay between direct tool invocations and sub-agent delegation, ensuring that execution complexity is effectively encapsulated while maintaining the flexibility to address diverse problem constraints.

3.2 Memory

The primary challenge in historical information compression is optimizing the granularity of salient information—minimizing loss from over-compression while maintaining a high-density information structure. To address this, we propose a working-memory management method comprising two core components: **Structured Memory Generation**, which handles the construction and dynamic updating of structured memory states, and **Dynamic Context Management**, which is responsible for dynamically and directly adjusting the context based on structured memory, providing the LLM with the most valuable and information-dense input.

3.2.1 Structured Memory Generation

We propose a structured memory architecture that prioritizes high-value history information along three dimensions: **sub-goal** (planning objectives), **tools-log** (tool invocation strategies), and **summary** (summary of information highly relevant to the query). Unlike linear history, our mechanism achieves high-dimensional abstraction by partitioning the agent's trajectory into semantically cohesive units based on sub-goals.

Memory Unit Structure Specifically, each memory unit is defined as a 4-tuple:

$$m_i = (\mathcal{R}_i, g_i, \mathcal{T}i, s_i) \tag{1}$$

where:

- $\mathcal{R}_i$ is the list of global round index contributing to the sub-goal.
- g_i is the semantic description of the current sub-goal, providing short-term guidance.
- $\mathcal{T}_i$ is the list of persistent tool-use log, consisting of tool names, parameters, and execution results.
- s_i is an incremental summary of key information extracted during task execution.

The complete memory list $\mathcal{M} = [m_1, m_2, \dots, m_n]$ represents a macro-level planning pathway that drives the agent toward task completion.

Dynamic Folding and Adding Mechanism An end-to-end memory model $\mathcal{F}_{\text{mem}}$ is introduced to manage transitions between sub-goals. At each interaction round t, $\mathcal{F}_{\text{mem}}$ processes the latest main-agent's response a_t, action resulting observation o_t and the latest memory unit $m_n (t > 1)$ to produce a binary indicator δ_{fold} and an updated unit m_{out}:

$$(m_{\text{out}}, \delta_{\text{fold}}) = \mathcal{F}_{\text{mem}}(a_t, o_t, m_n) \tag{2}$$

When $\delta_{\text{fold}} = 1$, it indicates the sub-goal of the current round is consistent with g_n. In this case, m_{out} as the latest updated memory unit, will directly replace the last element of the existing memory list:

$$\mathcal{M} \leftarrow \mathcal{M}_{1:n-1} \oplus [m_{\text{out}}] \tag{3}$$

where m_{out} incorporates the latest round index, the latest updated tool log τ_t and the latest updated summary s_n.

When $\delta_{\text{fold}} = 0$, it indicates that the sub-goal of the current round differs from the existing latest memory unit. In this case, m_{out} as a newly constructed memory unit will be appended to the memory list:

$$\mathcal{M} \leftarrow \mathcal{M} \oplus [m_{\text{out}}] \tag{4}$$

where m_{out} is constructed using the new sub-goal g_t, initial tool log τ_t and initial summary s_n.

This mechanism effectively compresses execution trajectories by aggregating multi-round interactions into single intentional units, significantly reducing context redundancy while preserving a high-fidelity history of the decision-making process.

3.2.2 Dynamic Context Management

Leveraging the structured memory $\mathcal{M}$, we propose a dynamic context management strategy to mitigate the linear context growth typical of the ReAct paradigm. By adaptively switching between fine-grained execution traces and compressed memory units, our approach ensures information integrity while optimizing efficiency.

Adaptive Context Construction Let Q be the user query and $|\mathcal{R}_n|$ be the count of the current memory unit. The context $\mathcal{C}_t$ can be constructed in the following two formats alternatively:

$$\mathcal{C}_t = \begin{cases} \mathcal{C}_{t-1} \oplus [r_t, o_t] & \text{if } |\mathcal{R}_n| > 1 \\ (Q, \mathcal{M}_{1:n-1}) \oplus [r_t, o_t] & \text{if } |\mathcal{R}_n| = 1 \end{cases} \tag{5}$$

When $|\mathcal{R}_n| > 1$, the agent is actively pursuing the current sub-goal g_n. We retain the incremental ReAct format, appending the latest response r_t and observation o_t to $\mathcal{C}_{t-1}$. This preserves the fine-grained execution trace necessary for tracking local dependencies within a sub-goal.

When $|\mathcal{R}_n| = 1$, it signals that the agent is starting a new sub-goal. To prevent redundant contextual information, we perform a compression reset, replacing historical round-by-round in model ReAct with serialized folded memory of all completed sub-goals $\mathcal{M}_{1:n-1}$.

Efficiency and Complexity Our memory management shifts the context complexity from $\mathcal{O}(t)$ total rounds to $\mathcal{O}(n)$ sub-goals. By triggering compression only upon sub-goal completion ($|\mathcal{R}_n| = 1$), the mechanism maintains a balance between high-level task awareness and low-level operational detail, ensuring that cross-sub-goal information is retained without incurring the redundancy of linear history.

3.3 Atomic Capability Pool

To construct a robust generalist assistant capable of navigating complex real-world workflows, we designed the atomic capability pool as a hierarchical system comprising two layers: *(i) Specialized Sub-agents*, which encapsulate high-level reasoning and multi-step planning for specific domains (GUI interaction and Data Analysis); and *(ii) Basic Tools* (detailed in Appendix C), which provide fundamental primitives for information retrieval and execution.

3.3.1 Browser-Use GUI Agent

To better align with our actual system implementation, we refer to this specialized sub-agent as the Browser-Use GUI Agent. Its interaction environment is restricted to a web browser, and it completes web-based subtasks through a closed loop of "observe–act–feedback", including searching, clicking, scrolling, typing, and content extraction on dynamic webpages. Unlike text-only fetching pipelines, this agent explicitly models *interactive elements* and *multi-tab states*, enabling robust information acquisition in scenarios that require real interaction (e.g., pagination, collapsible sections, login pop-ups, and lazy-loaded content).

Problem Formulation (POMDP). We model the interaction of the Browser-Use GUI Agent as a partially observable sequential decision process (POMDP), represented by the tuple

$$\mathcal{E} = \langle \mathcal{U}, \mathcal{A}, \mathcal{S}, \mathcal{O}, T \rangle, \tag{6}$$

where $\mathcal{U}$ denotes the task space (including user instructions and sub-goals), $\mathcal{A}$ the action space, $\mathcal{S}$ the underlying environment state space (not fully observable to the agent), $\mathcal{O}$ the observation space, and $T : \mathcal{S} \times \mathcal{A} \to \mathcal{P}(\mathcal{S})$ the state transition function.

State and Observation. In the browser setting, we define the true state $s_t \in \mathcal{S}$ as the complete information of the current browser environment. It includes not only structured web information (the full DOM tree, page contents of all tabs, the URL and navigation history, scroll and focus states, dynamic scripts and network request states, etc.), but also complete visual information (pixel-level rendering, layout and styles, and content outside the current window, including collapsed regions and visual content that becomes visible only after scrolling). This true state is not fully observable to the agent.

The observation $o_t \in \mathcal{O}$ obtained by the agent at step t consists of three parts:

$$o_t = (c_t, b_t, x_t), \tag{7}$$

where c_t is the textual context required to solve the task (e.g., the user query, the current sub-goal, key historical findings, and the previous action with its returned results), which contains no image information; b_t is a structured browser-state extracted from the currently visible window (i.e., the viewport), including the URL, title, tab list, scroll information, and interactive element indices with their corresponding DOM snippets limited to the current viewport; and x_t is a screenshot of the currently opened page (full-page rendering). Since b_t only covers the visible viewport (content outside the window cannot be directly "seen"), the process is inherently partially observable.

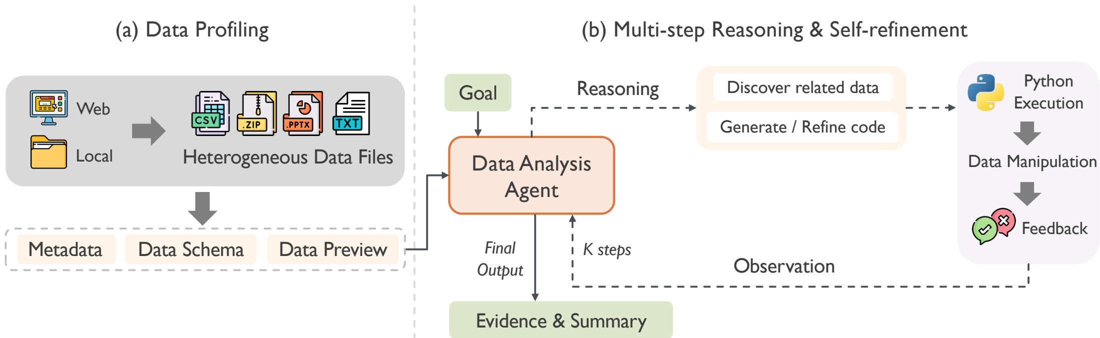


Figure 2: Illustration of Data Analysis Agent.

Importantly, in our implementation, the screenshot x_t is provided as ephemeral multi-modal input only for the current step, and is not serialized into the textual context history, preventing context explosion in long-horizon interactions.

Parameterized Discrete Actions (Tool-call). We represent each browser operation as a parameterized action

$$a_t = (\alpha_t, \theta_t) \in \mathcal{A}, \tag{8}$$

where α_t specifies the action type (e.g., `web_search`, `pdf_to_markdown`, `go_to_url`, `click_element`, `input_text`, `scroll_down/up`, `extract_content`, `open/switch/close_tab`, `terminate`), and θ_t contains structured parameters (e.g., URL, element index, input text, scroll pixels). In particular, for `pdf_to_markdown`, we adopt a paging mechanism to read long PDF documents incrementally: the converted Markdown is retrieved in pages (using an offset and a maximum character budget per page), rather than injected in full, which prevents context explosion in long-horizon interactions. In implementation, we enforce a “one tool per turn” constraint: at most one tool-call is executed per step, decomposing complex web interaction into a multi-round sequence of atomic decisions.

Policy. Due to partial observability, the policy π selects the next action based on the interaction history. Let

$$h_t = (u, o_1, a_1, \ldots, o_t) \tag{9}$$

denote the interaction history up to step t (including the task $u \in \mathcal{U}$, the observation sequence, and executed actions). The policy is defined as

$$\pi(a_t \mid h_t) : \Delta(\mathcal{H}) \to \mathcal{A}, \tag{10}$$

where $\mathcal{H}$ is the history space and $\Delta(\mathcal{H})$ denotes the probability simplex over it. The environment transitions according to T as $s_{t+1} \sim T(s_t, a_t)$ and produces the next observation o_{t+1}; the interaction terminates when `terminate` is selected or when a maximum step budget is reached.

3.3.2 Data Analysis Agent

The Data Analysis Agent is designed to automate complex data tasks by leveraging a multi-step reasoning process and generating executable code to perform intricate data processing, refine execution outputs, and generate evidence from heterogeneous sources. As illustrated in Figure 2, the workflow is structured into two primary phases: (a) Data Profiling, which yields comprehensive descriptions and previews of the input data, and (b) an iterative loop of multi-step reasoning and self-refinement to derive answers.

Data Profiling. The initial phase focuses on the comprehensive profiling of heterogeneous data files originating from web or local storage. The agent is engineered to handle a wide array of formats, including structured data, unstructured text, and compressed archives. Upon receiving files, the agent employs a routing mechanism to invoke appropriate tools—such as Pandas for tabular data or specialized parsers for documents—to extract essential information. This process yields a structured description comprising three key components: *(i) Metadata* (e.g., file name, size, type), *(ii) Data Schema* (e.g., dimensions, column names), and *(iii) Data Preview* (e.g., content snippets or full content, statistical summaries). This profiling step transforms raw files into a standardized context ready for analysis.

Multi-step Reasoning and Self-refinement. Building upon the meticulously compiled data profiles, the agent enters an iterative loop of reasoning and refinement driven by the user’s goal. This phase integrates the following sequential processes: *(i) Reasoning and Code Generation:* Based on the profiled context and current goal, the agent reasons to confirm file relevance, discover related data sections, and generate executable Python scripts to manipulate the data. *(ii) Execution and Feedback:* The generated code is executed within a secure sandbox, where the system captures the execution output or error message as feedback. *(iii) Observation and Self-refinement:* The agent assesses this feedback; if an error occurs, it triggers self-refinement to revise the code, whereas a successful observation informs the next reasoning step. This cycle continues until the objective is met, at which point the agent synthesizes the findings to produce the final output, which presents the supporting evidence and a final, coherent summary.

3.4 Adaptive Interrupt and Self-Correction Mechanism

In complex, long-horizon reasoning tasks, agents often succumb to cognitive inertia, where they persist in invalid behaviors despite execution failures. Common symptoms include syntactic errors (e.g., malformed tool calls) and semantic stagnation (e.g., repetitive outputs or recursive loops), which prevent the agent from progressing toward the solution.

To mitigate this, we introduce a Supervisor module equipped with an *Adaptive Interrupt and Self-Correction Mechanism* into the framework. Unlike rigid reflection schedules that might interfere with valid reasoning chains, the Supervisor continuously evaluates the agent's trajectory for anomalies. Upon detecting signals of failure or stagnation, it triggers a preemptive interrupt, forcibly transitioning the agent from an automated *Acting Mode* to a high-level *Reflective Mode*. In this state, the system executes a three-stage recovery protocol:

- **Anomaly Diagnosis:** The agent is prompted to critically analyze the execution history to pinpoint the root cause of failure.
- **Trajectory Pruning:** To prevent context pollution, the system explicitly prunes the recent invalid interaction traces from the context window. This ensures that the agent's future decisions are not biased by the failed attempts.
- **Re-generation:** The agent synthesizes an alternative output—such as a revised plan or a corrected conclusion—effectively breaking the local loop and guiding the workflow back to a valid solution path.

4 Experiments

Experimental Setup. In all experiments, we maintained consistent hyperparameters across comparable settings, setting the temperature to 1.0 and top-p to 0.95. The time allowance for each question is limited to 1.5 hours, and the maximum number of tool calls is capped at 75. All results were obtained in January 2026. Unless otherwise specified, all models employed within the framework are based on Gemini-3-pro to ensure consistent model capability across ablation studies. The maximum number of steps for specialized sub-agents is set to 10.

Benchmarks. We adhere to the official evaluation protocols for each benchmark. Our evaluation encompasses: (1) GAIA[30]; (2) BrowseComp[31]; (3) BrowseComp-ZH[32]; and (4) Humanity's Last Exam[33]. Detailed evaluation settings are provided in Appendix B.

Evaluation Metrics. The primary metric is *Pass@N*, defined as the probability that at least one correct solution appears among N independent executions. Unless otherwise specified, we report the *Pass@1* score, reflecting the model's ability to produce a correct answer in a single attempt.

Baselines. Our comparative analysis involves baselines categorized into three groups: (1) *General LLMs utilizing ReAct and basic tools*, including GLM-4.7[34], Kimi-K2 thinking[35], DeepSeek-V3.2[36], MiniMax M2[37], Claude Sonnet 4.5, GPT-5 High, OpenAI-o3 and Gemini 3 Pro; (2) *Open-source Agent Frameworks* such as DeepAgent[38], OAgent[39], MiroFlow[40], AgentFold[41], Tongyi DeepResearch[8], and WebResearcher[42]; and (3) *Closed-source Agent Frameworks* including Kimi-Researcher[6], Gemini Deep Research[5], and OpenAI Deep Research[4]. These selections cover a wide range of existing capabilities, establishing a solid foundation for benchmarking the performance of the Yunque DeepResearch framework.

4.1 Main Results

Table 1 presents a comprehensive comparison of Yunque DeepResearch against state-of-the-art baselines across four demanding benchmarks. Our framework demonstrates superior performance, achieving the highest scores on BrowseComp (62.5), BrowseComp-ZH (75.9), and Humanity's Last Exam (51.7), while securing a competitive second place on GAIA (78.6).

Comparative analysis reveals two key insights. First, Yunque DeepResearch significantly enhances the capabilities of base models. For instance, it boosts Gemini 3 Pro's performance by +10.0 on BrowseComp and +4.8 on GAIA compared to the standard ReAct baseline, demonstrating that our agentic design effectively unlocks the underlying LLM's potential. Second, our framework exhibits exceptional competitiveness against existing agentic systems; it consistently outperforms both open-source and closed-source baselines, validating its effectiveness in open-ended research.

Table 1: Performance comparison on various agentic benchmarks. **Bold** and underlined indicate the best and second-best results, respectively. Results marked with * are sourced from official technical reports.

Benchmarks	**Browse Comp**	**Browse Comp-ZH**	**GAIA**	**Humanity's Last Exam**	**Model Family**
General LLMs + ReAct + Basic Tools					
GLM 4.7	52.0*	66.6*	–	42.8*	–
Kimi K2 Thinking	<u>60.2</u>*	62.3*	69.9	44.9*	–
DeepSeek-V3.2	51.4*	65.0*	70.9	40.8*	–
MiniMax-M2	44.0*	–	75.7*	–	–
Claude-4.5-Sonnet	24.1*	42.4*	71.2*	32.0*	–
GPT-5 High	54.9*	63.0*	76.4*	35.2*	–
OpenAI-o3	49.7*	58.1*	70.5*	20.2*	–
Gemini 3 Pro	52.5	<u>67.1</u>	73.8	<u>45.8</u>*	–
Open-source Agent Frameworks					
DeepAgent	–	–	58.3*	20.2*	QwQ
Tongyi DeepResearch	43.4*	46.7*	70.9*	32.9*	Qwen
AgentFold	36.2*	47.3*	67.0*	–	Qwen
WebResearcher (w/o TTS)	37.3*	45.2*	72.8*	28.8*	Qwen
OAgent	22.2*	–	66.7*	15.4*	Claude
MiroFlow	33.2*	47.1*	**82.4***	27.2*	GPT
Close-source Agent Frameworks					
Gemini DeepResearch	–	–	–	26.9*	Gemini
Kimi Researcher	–	–	–	26.9*	Kimi
OpenAI DeepResearch	51.5*	42.9*	67.4*	26.6*	GPT
Yunque DeepResearch	**62.5**	**75.9**	<u>78.6</u>	**51.7**	Gemini

4.2 Detailed Analysis

Generalizability Across Backbones. To verify that the performance gains stem from our framework design rather than solely the capability of the underlying model, we evaluated Yunque DeepResearch using different backbone models: DeepSeek-V3.2, Kimi K2 Thinking, and Gemini 3 Pro. As shown in Table 2, our framework consistently enhances performance across diverse model families. Specifically, DeepSeek-V3.2 and Gemini 3 Pro exhibit robust improvements on all benchmarks, achieving peak gains of +10.1 (on BrowseComp-ZH) and +10.0 (on BrowseComp), respectively. While Kimi K2 Thinking shows a marginal fluctuation on BrowseComp-ZH, it secures a substantial +4.9 improvement on GAIA. We note that we selected these representative backbones based on their robust instruction-following capabilities, as precise tool invocation is a prerequisite for our framework that not all models currently satisfy. Collectively, these results confirm that Yunque DeepResearch is a model-agnostic solution that universally enhances problem-solving abilities.

Efficacy of Modular Design. We conducted an ablation study (Table 3) to assess the contribution of each component within Yunque DeepResearch. The results validate the necessity of our modular design:

- **Impact of Memory.** Removing the memory module results in the sharpest performance drop on browsing tasks (-10.4 on BrowseComp and -7.4 on BrowseComp-ZH). Since these benchmarks necessitate long-horizon information seeking and multi-hop reasoning, this decline underscores the critical role of our memory mechanism in mitigating the high noise ratio inherent in web browsing. While standard ReAct paradigms often suffer from context overflow and "lost-in-the-middle" phenomena—where valid information is submerged by

Table 2: Evaluation of our framework applied to different backbone models. Results marked with * are sourced from official technical reports.

Model	Mode	GAIA	BrowseComp-ZH	BrowseComp	HLE
DeepSeek-V3.2	ReAct	70.9	65.0*	51.4*	40.8*
	Ours	76.7	75.1	59.5	46.6
	Gap	(+5.8)	(+10.1)	(+8.1)	(+5.8)
Kimi K2 Thinking	ReAct	69.9	62.3*	–	44.9*
	Ours	74.8	61.9	–	–
	Gap	(+4.9)	(-0.4)	–	–
Gemini 3 Pro	ReAct	73.8	67.1	52.5	45.8*
	Ours	78.6	75.9	62.5	51.7
	Gap	(+4.8)	(+8.8)	(+10.0)	(+5.9)

invalid attempts—our memory management preserves information density by partitioning the agent's trajectory into semantically cohesive units via sub-goals.

- **Impact of Supervisor.** The Supervisor acts as a critical safeguard for complex task execution, with its removal precipitating significant performance declines across GAIA (-8.7), BrowseComp-ZH (-10.5), BrowseComp (-4.4), and HLE (-1.2). This module is essential for mitigating execution fragility in rigorous environments. Instead of allowing the agent to spiral into failure loops caused by malformed tool calls or ineffective actions, the Supervisor enforces a high-level orchestration layer that actively monitors for failure signals. By intercepting these errors and clearing invalid interaction traces from the context window, it prevents "error accumulation" and ensures the agent's reasoning process remains unpolluted. This mechanism guarantees that subsequent attempts are not biased by previous failures, thereby maintaining a clean and valid solution path during long-horizon reasoning.
- **Impact of Specialized Agents.** We prioritize the GAIA benchmark for this analysis, as its heterogeneous task composition—spanning file processing, data analysis, and open-world browsing—imposes a unique demand for domain-specific competencies. The ablation of the Browser-Use GUI Agent and Data Analysis Agent resulted in notable performance regressions (-6.8 and -2.9, respectively). This degradation corroborates the hypothesis that robust "general" assistance emerges from the orchestration of specialized atomic capabilities. These specialized modules provide the precise, low-level execution skills that monolithic models often lack when confronting the multimodal diversity of real-world tasks.

Table 3: Ablation study assessing the impact of individual components.

Method	GAIA	BrowseComp	BrowseComp-ZH	HLE
Yunque DeepResearch	**78.6**	**62.5**	**75.9**	**51.7**
w/o Memory	77.7	52.1	68.5	51.7
	(-0.9)	(-10.4)	(-7.4)	–
w/o Supervisor	69.9	58.1	65.4	50.5
	(-8.7)	(-4.4)	(-10.5)	(-1.2)
w/o Browser-Use GUI Agent	71.8	61.7	–	–
	(-6.8)	(-0.8)	–	–
w/o Data Analysis Agent	75.7	–	–	51.7
	(-2.9)	–	–	–

5 Limitations and Future Work

While our framework demonstrates promising results, we acknowledge several limitations that outline directions for future research. First, regarding the evaluation scope, although our main experiments encompass a diverse set of benchmarks, the ablation studies for specialized sub-agents are primarily anchored on GAIA. While GAIA serves as a robust testbed for generalist capabilities, it may not fully capture the nuances of specialized domains. Second, we have not yet conducted a systematic analysis of token consumption and inference latency. Although we explicitly designed sub-agents as lightweight workflows—characterized by concise instructions, finite execution horizons, and reduced context dependency—our empirical observations suggest that total execution time remains heavily dependent on the reasoning capability of the underlying foundation model.

In future work, we aim to extend the assessment of our specialized agents to include domain-specific benchmarks, such as DSBench[43] for comprehensive data analysis assessment and OSWorld[44] for complex, interactive environment navigation, thereby rigorously testing their performance boundaries. Furthermore, we plan to explore post-training strategies for atomic capabilities to develop smaller and more efficient modules, thereby reducing latency and computational costs.

6 Conclusion

We presented Yunque DeepResearch, a hierarchical, modular, and robust multi-agent framework designed to address the critical challenges of cognitive overload in long-horizon tasks, systemic fragility in autonomous execution, and the lack of modular extensibility in rigid architectures. Extensive empirical evaluations demonstrate that Yunque DeepResearch achieves state-of-the-art results on challenging benchmarks, including GAIA, BrowseComp, BrowseComp-ZH, and Humanity's Last Exam. We hope that our open-source contributions will provide a solid foundation for future research on collaborative agentic systems.

Appendix

A Case Study

In this section, we select one question from the BrowseComp dataset as an example of a long-horizon task to illustrate our memory generation process in detail. Additionally, we select a challenging information retrieval case from the GAIA dataset as an example, showcasing our agent's problem-solving workflow to illustrate the superiority of proposed hierarchical framework.

Example 1: Memory The following question selected is a typical cross-domain and multi-step reasoning challenge. Its reasoning path involves multi-level logical progression, where the failure to verify any minor detail can lead to the systemic collapse of the entire reasoning chain. Our dynamic memory management mechanism achieves a structured evolution of memory storage by capturing the real-time strategies of the main agent. Each memory unit encapsulates key information consisting of 'Goal-Action Process-Outcome'. Based on sub-goals, these memory units form a continuous reasoning chain that reflects the agent's systematic 'Investigation-Reasoning-Verification' workflow.

Case Memory Generation for Example 1 in BrowseComp

Question: There's a thesis submitted between 2020 and 2023, inclusive, for a Doctor of Philosophy degree at a university established between January 1st, 1980, and December 31, 1990, inclusive. The author dedicated the thesis to their children and the thesis is related to dating apps. In its acknowledgment, the author mentioned about their committed relationship coming to an end and starting a podcast. The author started this podcast with someone they originally met at a film event. What's the name of the podcast?

[Memory Unit 1]: Round 1

sub-goal: Investigate Piia Myyry as the potential author and identify the podcast name

tools-log:

```
[{
    "tool": "search",
    "args": {
        "query": ["PhD thesis \"dating apps\" dedication \"children\" 2020..2023", "PhD
    thesis \"dating apps\" acknowledgment \"podcast\" \"film event\"", "PhD thesis \"dating
    apps\" acknowledgment \"relationship\" \"podcast\"", "PhD thesis \"dating apps\"
    dedicated to my children"]
    }",
    "status": "success"
}]
```

summary: The agent executed searches for specific phrases in thesis acknowledgments. While the agent's thought process initially focused on Lisa Portolan, the tool output surfaced a strong lead for 'Piia Myyry', whose thesis is associated with the phrases 'met at a film event' and 'started a podcast' in the context of dating apps.

[Memory Unit 2]: Round 2~3

sub-goal: Verify Lisa Portolan as the author and identify the specific podcast

tools-log:

```
[{
    "tool": "search",
    "args": {
        "query": ["Lisa Portolan \"Slow Love\" podcast met co-host film event", "Lisa
    Portolan thesis acknowledgments \"relationship\" \"podcast\"", "Lisa Portolan \"Slow
    Love\" podcast \"breakup\"", "Lisa Portolan PhD thesis dedication children"]
    },
    "status": "success"
}]
```

Summary: Search results confirm she has a PhD from Western Sydney University (established 1989) focusing on dating apps and has a podcast called 'Slow Love'. The next steps focus on finding the full text or specific acknowledgments of her thesis to confirm the personal details and the podcast name.

[Memory Unit 3]: Round 4~5

sub-goal: Identify the podcast name by searching for specific phrases from the thesis acknowledgment

tools-log:


```
[
    {
        "tool": "search",
        "args": {
            "query": ["\"started a podcast with whom I first met at a film event\"", "\"my
    committed relationship came to an end\" podcast thesis", "Lisa Portolan thesis
    acknowledgments \"film event\"", "Lisa Portolan \"Slow Love\" podcast co-host film event
    ", "PhD thesis dating apps \"dedicated to my children\""]
        },
        "status": "success"
    },
    {
        "tool": "search",
        "args": {
            "query": ["\"started a podcast with whom I first met at a film event\"", "\"my
    committed relationship came to an end\" thesis", "\"A Lifestyle Gap\" PhD thesis dating
    apps", "Lisa Portolan thesis \"film event\"", "podcast co-host \"met at a film event\""]
        },
        "status": "success"
    }
]
```


summary: The agent confirmed the unique text string: 'my committed relationship came to an end and i started a podcast with whom i first met at a film event'. Piia Myyry and 'A Lifestyle Gap' appear in snippets but may be noise; the agent suspects the author is Lisa Portolan.

[Memory Unit 4]: Round 6

sub-goal: Verify the thesis title 'A Lifestyle Gap' and the podcast name by confirming the author's identity and the 'film event' origin story

tools-log:


```
[{
    "tool": "search",
    "args": {
        "query": ["\"A Lifestyle Gap\" PhD thesis", "\"A Lifestyle Gap\" Lisa Portolan", "\"
    Slow Love\" podcast Lisa Portolan film event", "Lisa Portolan podcast \"met at a film
    event\"", "\"my committed relationship came to an end\" podcast"]
    },
    "status": "success"
}]
```


summary: Search results strongly suggest the author is Lisa Portolan. The text 'my committed relationship came to an end and i started a podcast with whom i first met at a film event' appears in snippets linked to her.

[Memory Unit 5]: Round 7

sub-goal: Confirm Lisa Portolan as the author, verify the thesis title, and identify the podcast name

tools-log:


```
[{
    "tool": "search",
    "args": {
```

```
        "query": ["\"A Lifestyle Gap\" thesis", "\"A Lifestyle Gap\" dating apps", "\"
    started a podcast\" \"film event\" thesis", "site:edu \"dedicated to my children\" \"
    committed relationship\" podcast", "\"my committed relationship came to an end\" thesis
    acknowledgment"]
    },
    "status": "success"
}]
```

summary: Scraper results contain the mashed-up snippet linking 'A Lifestyle Gap', 'dedicated to my children', and the 'film event' story. The author is likely Lisa Portolan (Western Sydney University, est. 1989).

[Memory Unit 6]: Round 8~9

sub-goal: Confirm the podcast name (likely 'Slow Love') by verifying the 'met at a film event' story in Lisa Portolan's thesis acknowledgments

tools-log:

```
[
    {
        "tool": "search",
        "args": {
            "query": ["Lisa Portolan thesis acknowledgment \"film event\" podcast", "\"Slow
    Love\" podcast Lisa Portolan co-host", "Lisa Portolan \"committed relationship came to
    an end\"", "Did Lisa Portolan start a podcast with someone she met at a film event?", "
    thesis dedicated to my children dating apps podcast film event"]
        }",
        "status": "success"
    },
    {
        "tool": "search",
        "args": {
            "query": ["Lisa Portolan Jodi McAlister podcast", "Lisa Portolan \"Slow Love\"
    podcast co-host", "\"Slow Love\" podcast Lisa Portolan met at a film event", "Lisa
    Portolan podcast \"committed relationship came to an end\"", "Lisa Portolan thesis
    acknowledgements text", "Lisa Portolan \"The 40 Year Old Wino\"", "Lisa Portolan \"
    Dating in the Digital Age\" podcast"]
        },
        "status": "success"
    }
]
```

summary: Lisa Portolan is confirmed as the author. Search results explicitly link the phrase 'met at a film event' and 'started a podcast' to her and the podcast 'Slow Love'.

[Memory Unit 7]: Round 10

sub-goal: Verify the podcast name and the co-host meeting story within Lisa Portolan's thesis acknowledgments

tools-log:

```
[
    {
        "tool": "search",
        "args": {
            "query": ["Lisa Portolan thesis acknowledgment \"committed relationship came to
    an end\"", "\"Slow Love\" podcast Lisa Portolan started with", "Lisa Portolan podcast \"
    met at a film event\"", "Did Lisa Portolan meet her podcast co-host at a film event?", "
    Lisa Portolan thesis \"dedicated to my children\""]
        },
        "status": "success"
    }
]
```

summary: The snippet 'Slow Love Podcast Lisa Portolan Co-host Met At Film Event' appearing in SEO results strongly suggests 'Slow Love' is the correct answer.

[Memory Unit 8]: Round 11~12

sub-goal: Identify the podcast name and author by verifying the specific acknowledgment phrase

tools-log:


```
[
    {
        "tool": "search",
        "args": {
            "query": ["\"started a podcast with whom I first met at a film event\"", "\"my
    committed relationship came to an end\" thesis", "\"A Lifestyle Gap\" PhD thesis dating
    apps", "Lisa Portolan thesis \"film event\"", "podcast co-host \"met at a film event\""]
        },
        "status": "success"
    },
    {
        "tool": "google_scholar",
        "args": {
            "query": ["\"A Lifestyle Gap\" PhD thesis", "\"A Lifestyle Gap\" dating apps",
    "\"started a podcast\" \"film event\" thesis", "site:edu \"dedicated to my children\" \"
    committed relationship\" podcast", "\"my committed relationship came to an end\" thesis
    acknowledgment"]
        },
        "status": "success"
    }
]
```


summary: Google Scholar returned no results for 'A Lifestyle Gap', indicating it's likely a scraper artifact. Lisa Portolan remains the primary candidate (WSU est. 1989, dating apps topic, 'Slow Love' podcast).

Example 2: Framework This example demonstrates the framework's capability to handle complex real-world from the GAIA benchmark. The trajectory highlights the seamless orchestration between specialized sub-agents, facilitated by structured memory management: the Browser-Use Agent precisely locates the dataset URL, while the Data Analysis Agent downloads the CSV file and executes Python code to perform rigorous data cleaning and verification, overcoming the calculation errors and context limitations common in standard LLMs.

Case Trajectory for Example 2 in GAIA

Question: According to the World Bank, which countries had gross savings of over 35% of GDP for every year in the period 2001-2010? Give your answer as a comma-separated list of countries in alphabetical order. Use the countries most common names in english when answering.

Step 1: Planning & Information Seeking

<think>
We need to find data from World Bank on gross savings (% of GDP) for each country for each year 2001-2010 inclusive. Then find those countries where gross savings >35% for each year in that period.
</think>

<tool_call>
```
{"name": "search", "arguments": {"query": ["World Bank gross savings % of GDP by
country 2001-2010 data"]}}
```
</tool_call>

<tool_response>
A Google search for 'World Bank gross savings % of GDP by country 2001-2010 data' found 10 results:

1. [Gross savings (% of GDP)] (https://data.worldbank.org/indicator/NY.GNS.ICTR.ZS)
Gross savings (% of GDP) Country official statistics, National Statistical Organizations and/or Central Banks; National Accounts data files, Organisation for ...

...
</tool_response>

<memory>
steps [1]:
{“sub_goal”: “Identify the World Bank indicator code for ’Gross savings (% of GDP)”, “tools_log”: ..., “summary”: “The agent searched for World Bank gross savings data and located the official indicator page, identifying the relevant indicator code as ‘NY.GNS.ICTR.ZS’.”}
</memory>

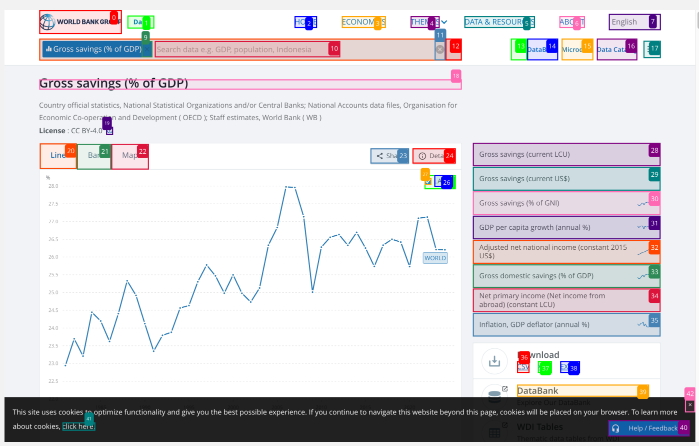
Step 2: Navigation & Extraction (Browser-Use Agent)

<think>
We need to find the list of countries where gross savings (% of GDP) >35% for each year 2001-2010 inclusive. The World Bank indicator for gross savings (% of GDP) is "NY.GNS.ICTR.ZS". Let’s click that.
</think>

<tool_call>
```
{"name": "visit", "arguments": {"url": ["https://data.worldbank.org/indicator/
NY.GNS.ICTR.ZS"], "goal": "Find the data for gross savings (% of GDP) for all
countries for years 2001-2010 and identify which countries have values >35% for
each year."}}
```
</tool_call>

↪ *Delegating task to* ***Browser-Use Agent****...*

<tool_response>
Screenshot:



Thinking trace:
I see the download options on the right side of the page under the “Download” header. There are links for CSV, XML, and EXCEL. I need to get the direct download link for the CSV or Excel file. I will check the ‘href‘ attribute of element [36] (CSV) which is ‘https://api.worldbank.org/v2/en/indicator/NY.GNS.ICTR.ZS?downloadformat=csv’. This looks like a direct download link. I will provide this link.

Summary:
The World Bank Open Data platform provides comprehensive data on ‘Gross savings (% of GDP)’ (indicator NY.GNS.ICTR.ZS) for all countries from 1960 to 2024, which includes the requested time period of 2001–2010. The data can be downloaded in CSV formats via the provided link (https://api.worldbank.org/v2/en/indicator/NY.GNS.ICTR.ZS?downloadformat=csv), enabling users to extract and analyze values for each country.
</tool_response>

Step 3: Data Processing (Data Analysis Agent)

<think>
The page indicates we can download CSV. Let's download the CSV file.
</think>

<tool_call>

```
{"name": "visit", "arguments": {"url": ["https://api.worldbank.org/v2/en/indicator/
NY.GNS.ICTR.ZS?downloadformat=csv"], "goal": "Download the CSV file containing
gross savings (% of GDP) data for all countries from 1960 to 2024."}}
```

</tool_call>

↪ *Delegating task to* ***Data Analysis Agent****...*

<tool_response>
Deep analysis for extracted file Metadata_Indicator_API_NY.GNS.ICTR.ZS_DS2_en_csv_v2_2962.csv in zip file:
<rational>
The provided file, 'Metadata_Indicator_API_NY.GNS.ICTR.ZS_DS2_en_csv_v2_2962.csv', is a metadata file documenting the definition and source of the indicator "Gross savings (% of GDP)". It contains descriptive text (e.g., "Savings is an amount that represent...") but does not contain any numerical time-series data or country-specific rows. To answer the ultimate question regarding which countries had gross savings of over 35% of GDP between 2001 and 2010, the primary data file (typically named 'API_NY.GNS.ICTR.ZS_DS2_en_csv_v2_... .csv' in World Bank datasets) containing the actual values for countries and years is required. This file is irrelevant for the calculation step.
</rational>

=======

Deep analysis for extracted file API_NY.GNS.ICTR.ZS_DS2_en_csv_v2_2962.csv in zip file:
<rational>
The user's goal is to identify countries with gross savings (% of GDP) strictly greater than 35% for every year in the period 2001-2010 using the provided CSV file.
I executed a Python script ("Path B") to process the file 'API_NY.GNS.ICTR.ZS_DS2_en_csv_v2_2962.csv'.
The script:
1. Read the CSV file, correctly skipping the metadata rows.
2. Filtered out non-country regional aggregates (e.g., "Arab World", "High income") to ensure only sovereign countries were listed.
3. Checked the condition 'Gross savings > 35.0' for each country for every year from 2001 through 2010.
4. Renamed countries to their common English names (e.g., "Brunei Darussalam" to "Brunei").
5. Output the resulting list alphabetically.
The code execution was successful and returned the list: 'Brunei, China, Morocco, Singapore'. These are the countries that met the specified criteria according to the data in the provided file.
</rational>

<evidence>
The code execution produced the following output: 'Brunei, China, Morocco, Singapore'
</evidence>

<summary>
According to the World Bank data in the provided file, the countries that had gross savings of over 35% of GDP for every year in the period 2001-2010 are Brunei, China, Morocco, and Singapore.
</summary>

=======

Deep analysis for extracted file Metadata_Country_API_NY.GNS.ICTR.ZS_DS2_en_csv_v2_2962.csv in zip file:
<rational>
The provided file 'Metadata_Country_API_NY.GNS.ICTR.ZS_DS2_en_csv_v2_2962.csv' contains metadata regarding countries and regional aggregates (e.g., Region, IncomeGroup, SpecialNotes) but does not contain the actual Gross Savings (% of GDP) time-series data. The values required to determine which countries had gross savings over 35% of GDP from 2001 to 2010 are missing from this file. The user likely intended to provide the primary data file (usually named 'API_NY.GNS.ICTR.ZS_DS2_en_csv_v2_2962.csv'). Consequently, this file is irrelevant for performing the specific data analysis requested in the ultimate question, although it would be

useful for filtering out non-country aggregates if the data file were present.
</rational>
</tool_response>

<memory>
steps [2, 3]:
{"sub_goal": "Download and process the 'Gross savings (% of GDP)' dataset", "tools_log": ..., "summary": "The agent visited the World Bank indicator page, confirmed the code 'NY.GNS.ICTR.ZS', and identified the CSV download URL. Subsequently, the agent successfully executed a self-contained Python script to download, extract, and process the World Bank dataset directly. The script filtered for entities with gross savings > 35% for every year from 2001 to 2010. The resulting list includes: Brunei, China, Morocco, Singapore. The agent is now ready to provide the final answer."}
</memory>

Step 4: Final Synthesis

<think>
I will output the answer as a comma-separated list of common names. Common names: Brunei, China, Morocco, Singapore. One final check: Is there any other country? My script filtered all countries in the CSV. The CSV contains all World Bank data. I am confident in the list: Brunei, China, Morocco, Singapore. Final Answer seems ready.
</think>

<answer>
Brunei, China, Morocco, Singapore
</answer>

B Evaluation Details

For the GAIA benchmark, we employ Qwen3-235b-a22b-2507 as the judge model, utilizing the text-only validation subset in accordance with the settings in [45]. For Humanity's Last Exam (HLE), we evaluate performance on the subset of 2,158 text-only questions. For BrowseComp, BrowseComp-ZH, and HLE, we utilize GPT-4o-2024-08-06 as the judge model. To ensure consistency and comparability, all evaluation prompts across these benchmarks adhere to their official protocols. We selected these four benchmarks to provide a comprehensive assessment of our framework's capabilities:

- **GAIA[30]** - A benchmark comprising real-world, challenging tasks designed to evaluate general AI assistants. We utilize the 103-sample text-only subset to assess capabilities such as multi-step reasoning, web browsing, and general tool-use proficiency.
- **BrowseComp[31]** - A simple yet challenging benchmark consisting of 1,266 questions designed to measure an agent's web browsing capabilities. It requires agents to persistently navigate the internet to locate hard-to-find and entangled information.
- **BrowseComp-ZH[32]** - A high-difficulty Chinese benchmark containing 289 multi-hop questions specifically tailored to retrieve information within the Chinese web ecosystem.
- **HLE[33]** - Humanity's Last Exam is an expert-curated benchmark featuring challenging questions that span a wide range of disciplines. It is designed to test expert-level performance on closed-ended, verifiable questions involving cutting-edge scientific knowledge.

C Details of Basic Tools

The framework is equipped with a core set of fundamental capabilities, serving as the primitives for both the Main Agent and the specialized sub-agents.

Search. To acquire external knowledge, the `Search` tool interfaces with search engines (e.g., Google, Google Scholar). It retrieves relevant URLs and snippets, enabling the agent to locate information sources dynamically.

Read & Parse. Information ingestion is handled by a hybrid strategy tailored to the source format: *(i) Web Content:* For web pages, the tool utilizes lightweight APIs (e.g., Jina Reader) to convert HTML into clean Markdown. This serves as a lightweight, high-efficiency complement to the Browser-Use GUI Agent. *(ii) File Content:* A unified `FileParser` handles diverse local formats.

Code Execution. For tasks requiring computation or logic verification, the `Code Execution` tool provides a secure, Docker-based Python sandbox. It supports file I/O and library imports, allowing the agent to solve mathematical problems, process data, and verify algorithmic logic in an isolated environment.

References

[1] Shunyu Yao, Jeffrey Zhao, Dian Yu, Nan Du, and Izhak Shafran. ReAct: Synergizing Reasoning and Acting in Language Models. *arXiv preprint arXiv:2210.03629*.

[2] Noah Shinn, Federico Cassano, Edward Berman, Ashwin Gopinath, and Karthik Narasimhan. Reflexion: Language Agents with Verbal Reinforcement Learning. *arXiv preprint arXiv:2303.11366*.

[3] Timo Schick, Jane Dwivedi-Yu, Roberto Dessì, Roberta Raileanu, and Maria Lomeli. Toolformer: Language Models Can Teach Themselves to Use Tools. *arXiv preprint arXiv:2302.04761*.

[4] OpenAI. Deep research system card. `https://cdn.openai.com/deep-research-system-card.pdf`.

[5] Gemini Team. Gemini Deep Research — your personal research assistant. `https://gemini.google/overview/deep-research/`.

[6] Kimi. Kimi-Researcher: End-to-End RL Training for Emerging Agentic Capabilities. `https://moonshotai.github.io/Kimi-Researcher/`.

[7] Chen Hu, Haikuo Du, Heng Wang, Lin Lin, and Mingrui Chen. Step-DeepResearch Technical Report. *arXiv preprint arXiv:2512.20491*.

[8] Tongyi DeepResearch Team, Baixuan Li, Bo Zhang, Dingchu Zhang, Fei Huang, Guangyu Li, Guoxin Chen, Huifeng Yin, Jialong Wu, Jingren Zhou, et al. Tongyi deepresearch technical report. *arXiv preprint arXiv:2510.24701*, 2025.

[9] Kuan Li, Zhongwang Zhang, Huifeng Yin, Liwen Zhang, and Litu Ou. WebSailor: Navigating Super-human Reasoning for Web Agent. *arXiv preprint arXiv:2507.02592*.

[10] Mert Cemri, Melissa Z. Pan, Shuyi Yang, Lakshya A. Agrawal, and Bhavya Chopra. Why Do Multi-Agent LLM Systems Fail? *arXiv preprint arXiv:2503.13657*.

[11] Xu Shen, Qi Zhang, Song Wang, Zhen Tan, and Xinyu Zhao. Metacognitive Self-Correction for Multi-Agent System via Prototype-Guided Next-Execution Reconstruction. *arXiv preprint arXiv:2510.14319*.

[12] Junyu Luo, Weizhi Zhang, Ye Yuan, Yusheng Zhao, and Junwei Yang. Large Language Model Agent: A Survey on Methodology, Applications and Challenges. *arXiv preprint arXiv:2503.21460v1*.

[13] OpenAI. Introducing deep research. `https://openai.com/index/introducing-deep-research/`, 2025.

[14] Google. Try deep research and our new experimental model in gemini, your ai assistant. `https://blog.google/products/gemini/google-gemini-deep-research/`, 2025.

[15] MiniMax. Minimax agent: Minimize effort, maximize intelligent. `https://agent.minimax.io/`, 2025.

[16] Wentao Zhang, Ce Cui, Yilei Zhao, Yang Liu, and Bo An. Agentorchestra: A hierarchical multi-agent framework for general-purpose task solving. *arXiv preprint arXiv:2506.12508*, 2025.

[17] Chengyue Yu, Siyuan Lu, Chenyi Zhuang, Dong Wang, Qintong Wu, Zongyue Li, Runsheng Gan, Chunfeng Wang, Siqi Hou, Gaochi Huang, Wenlong Yan, Lifeng Hong, Aohui Xue, Yanfeng Wang, Jinjie Gu, David Tsai, and Tao Lin. Aworld: Orchestrating the training recipe for agentic ai, 2025.

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

[19] He Zhu, Tianrui Qin, King Zhu, Heyuan Huang, Yeyi Guan, Jinxiang Xia, Yi Yao, Hanhao Li, Ningning Wang, Pai Liu, Tianhao Peng, Xin Gui, Xiaowan Li, Yuhui Liu, Yuchen Eleanor Jiang, Jun Wang, Changwang Zhang, Xiangru Tang, Ge Zhang, Jian Yang, Minghao Liu, Xitong Gao, Wangchunshu Zhou, and Jiaheng Liu. Oagents: An empirical study of building effective agents, 2025.

[20] Xixi Wu, Kuan Li, Yida Zhao, Liwen Zhang, Litu Ou, Huifeng Yin, Zhongwang Zhang, Xinmiao Yu, Dingchu Zhang, Yong Jiang, et al. Resum: Unlocking long-horizon search intelligence via context summarization. *arXiv preprint arXiv:2509.13313*, 2025.

[21] Zijian Zhou, Ao Qu, Zhaoxuan Wu, Sunghwan Kim, Alok Prakash, Daniela Rus, Jinhua Zhao, Bryan Kian Hsiang Low, and Paul Pu Liang. Mem1: Learning to synergize memory and reasoning for efficient long-horizon agents, 2025.

[22] Yuxiang Zhang, Jiangming Shu, Ye Ma, Xueyuan Lin, Shangxi Wu, and Jitao Sang. Memory as action: Autonomous context curation for long-horizon agentic tasks, 2025.

[23] Hongli Yu, Tinghong Chen, Jiangtao Feng, Jiangjie Chen, Weinan Dai, Qiying Yu, Ya-Qin Zhang, Wei-Ying Ma, Jingjing Liu, Mingxuan Wang, et al. Memagent: Reshaping long-context llm with multi-conv rl-based memory agent. *arXiv preprint arXiv:2507.02259*, 2025.

[24] Rui Ye, Zhongwang Zhang, Kuan Li, Huifeng Yin, Zhengwei Tao, Yida Zhao, Liangcai Su, Liwen Zhang, Zile Qiao, Xinyu Wang, Pengjun Xie, Fei Huang, Siheng Chen, Jingren Zhou, and Yong Jiang. AgentFold: Long-horizon web agents with proactive context management.

[25] Wenyi Hong, Weihan Wang, Qingsong Lv, Jiazheng Xu, Wenmeng Yu, Junhui Ji, Yan Wang, Zihan Wang, Yuxiao Dong, Ming Ding, and Jie Tang. Cogagent: A visual language model for gui agents, 2023.

[26] Anthropic. Introducing computer use, a new claude 3.5 sonnet, and claude 3.5 haiku. `https://www.anthropic.com/news/3-5-models-and-computer-use`, 2024.

[27] OpenAI. Data analysis with chatgpt. `https://help.openai.com/en/articles/8437071-data-analysis-with-chatgpt`, 2025.

[28] Sirui Hong, Yizhang Lin, Bang Liu, Bangbang Liu, Binhao Wu, Ceyao Zhang, Danyang Li, Jiaqi Chen, Jiayi Zhang, Jinlin Wang, et al. Data interpreter: An llm agent for data science. In *Findings of the Association for Computational Linguistics: ACL 2025*, pages 19796–19821, 2025.

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

[30] Grégoire Mialon, Clémentine Fourrier, Craig Swift, Thomas Wolf, and Yann LeCun. GAIA: A benchmark for General AI Assistants. *arXiv preprint arXiv:2311.12983*, 2023.

[31] Jason Wei, Zhiqing Sun, Spencer Papay, Scott McKinney, and Jeffrey Han. BrowseComp: A Simple Yet Challenging Benchmark for Browsing Agents. *arXiv preprint arXiv:2504.12516*, 2025.

[32] Peilin Zhou, Bruce Leon, Xiang Ying, Can Zhang, and Yifan Shao. BrowseComp-ZH: Benchmarking Web Browsing Ability of Large Language Models in Chinese. *arXiv preprint arXiv:2504.19314*, 2025.

[33] Long Phan, Alice Gatti, Ziwen Han, Nathaniel Li, and Josephina Hu. Humanity's Last Exam. *arXiv preprint arXiv:2501.14249*, 2025.

[34] GLM-4.5 Team, Aohan Zeng, Xin Lv, Qinkai Zheng, and Zhenyu Hou. GLM-4.5: Agentic, Reasoning, and Coding (ARC) Foundation Models. *arXiv preprint arXiv:2508.06471*.

[35] Kimi Team, Yifan Bai, Yiping Bao, Guanduo Chen, and Jiahao Chen. Kimi K2: Open Agentic Intelligence. *arXiv preprint arXiv:2507.20534*.

[36] DeepSeek-AI. DeepSeek-V3.2: Pushing the Frontier of Open Large Language Models.

[37] MiniMax-AI. MiniMax M2 & Agent: Ingenious in Simplicity. `https://www.minimax.io`.

[38] Xiaoxi Li, Wenxiang Jiao, Jiarui Jin, Guanting Dong, and Jiajie Jin. DeepAgent: A General Reasoning Agent with Scalable Toolsets. *arXiv preprint arXiv:2510.21618*.

[39] He Zhu, Tianrui Qin, King Zhu, Heyuan Huang, and Yeyi Guan. OAgents: An Empirical Study of Building Effective Agents. *arXiv preprint arXiv:2506.15741*.

[40] MiroMindAI. MiroFlow. `https://github.com/MiroMindAI/MiroFlow`.

[41] Rui Ye, Zhongwang Zhang, Kuan Li, Huifeng Yin, and Zhengwei Tao. AgentFold: Long-Horizon Web Agents with Proactive Context Management. *arXiv preprint arXiv:2510.24699*.

[42] Zile Qiao, Guoxin Chen, Xuanzhong Chen, Donglei Yu, and Wenbiao Yin. WebResearcher: Unleashing unbounded reasoning capability in Long-Horizon Agents. *arXiv preprint arXiv:2509.13309*.

[43] Liqiang Jing, Zhehui Huang, Xiaoyang Wang, Wenlin Yao, and Wenhao Yu. DSBench: How Far Are Data Science Agents from Becoming Data Science Experts? In *ICLR 2025*.

[44] Tianbao Xie, Danyang Zhang, Jixuan Chen, Xiaochuan Li, and Siheng Zhao. OSWorld: Benchmarking Multimodal Agents for Open-Ended Tasks in Real Computer Environments. *arXiv preprint arXiv:2404.07972*.

[45] Xiaoxi Li, Jiajie Jin, Guanting Dong, Hongjin Qian, and Yutao Zhu. WebThinker: Empowering Large Reasoning Models with Deep Research Capability. *arXiv preprint arXiv:2504.21776*, 2025.