Tongyi DeepResearch Technical Report

Tongyi DeepResearch Team*

Tongyi Lab , Alibaba Group

https://tongyi-agent.github.io/blog
https://github.com/Alibaba-NLP/DeepResearch
https://huggingface.co/Alibaba-NLP/Tongyi-DeepResearch-30B-A3B
https://www.modelscope.cn/models/iic/Tongyi-DeepResearch-30B-A3B

Abstract

We present **Tongyi DeepResearch**, an agentic large language model, which is specifically designed for long-horizon, deep information-seeking research tasks. To incentivize autonomous deep research agency, Tongyi DeepResearch is developed through an end-to-end training framework that combines agentic mid-training and agentic post-training, enabling scalable reasoning and information seeking across complex tasks. We design a highly scalable data synthesis pipeline that is fully automatic, without relying on costly human annotation, and empowers all training stages. By constructing customized environments for each stage, our system enables stable and consistent interactions throughout. Tongyi DeepResearch, featuring 30.5 billion total parameters, with only 3.3 billion activated per token, achieves state-of-the-art performance across a range of agentic deep research benchmarks, including Humanity's Last Exam, BrowseComp, BrowseComp-ZH, WebWalkerQA, xbench-DeepSearch, FRAMES and xbench-DeepSearch-2510. We open-source the model, framework, and complete solutions to empower the community.

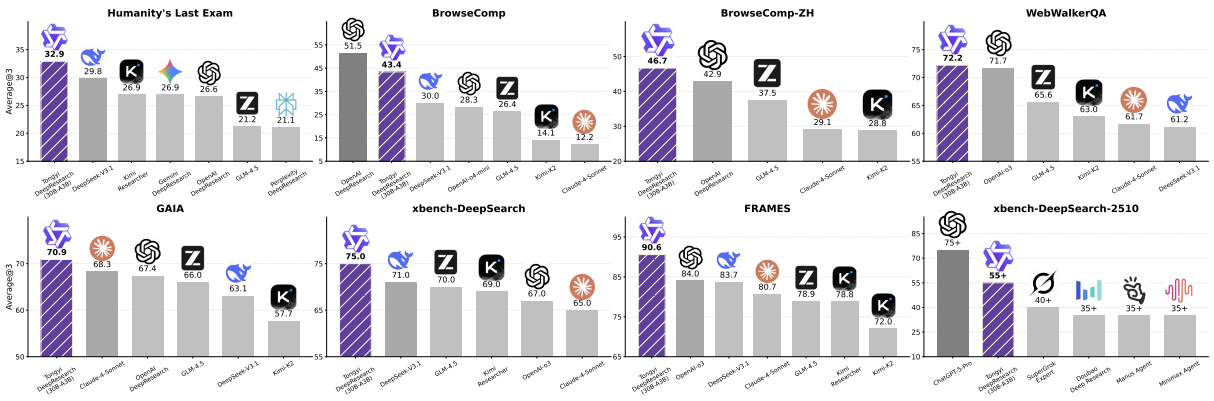


Figure 1: Benchmark performance of Tongyi DeepResearch.

*Full author list available in the Contributions section.

1 Introduction

As we advance toward Artificial General Intelligence (AGI), the emergence of Deep Research agents offers a promising paradigm for augmenting and potentially liberating human intellectual productivity. Deep research is a new agentic capability that autonomously conducts multi-step reasoning and information seeking on the internet for complex research tasks. It can be completed in tens of minutes, which would otherwise require several hours for a human (OpenAI, 2025a; Claude Team, 2025; Grok Team, 2025; Gemini Team, 2025). However, most deep research systems remain closed-source, and their intermediate research processes are inaccessible. While the community has made preliminary explorations in this area (Wu et al., 2025a; Li et al., 2025c; Tao et al., 2025), there is still a lack of a systematic methodology and publicly available models that can be fully open-sourced and shared across the community.

In this work, we introduce **Tongyi DeepResearch**, opening the era of open-source AI researchers. Our goal is to endow large language models (LLMs) with autonomous research capabilities agency, the ability to plan, search, reason, and synthesize knowledge across extended sequences of actions and diverse information sources.

Tongyi DeepResearch delivers several key advancements:

- We propose an **end-to-end agentic training paradigm** that unifies agentic mid-training and agentic post-training, forming a scalable foundation for deep reasoning and information-seeking behaviors. Agentic mid-training cultivates inherent agentic biases by exposing the model to large-scale, high-quality agentic data, serving as a progressive transition from pre-training to post-training stages. Agentic post-training further unlocks the model's potential via scalable multi-turn reinforcement learning on a strong base model. Together, they enable the model to gradually develop from basic interaction skills to advanced autonomous research behaviors.
- We design a **fully automated, highly scalable data synthesis pipeline** that eliminates human annotation while generating diverse, high-quality agent trajectories. We design stage-specific data synthesis strategies tailored to the objectives of each training phase, ensuring that every stage is supported by appropriately structured and targeted data. Synthetic data is highly scalable, fast to validate, and enables the construction of super-human-level datasets with stable distributions. It serves as an indispensable engine for agent training.
- We construct **stage-specific, customized environments** that rely on robust infrastructure to deliver consistent interactions for data synthesis across training stages. These environments allow the agent to engage in rich, specialized interactions that are tightly aligned with its developmental stage. They can take various forms, from prior world models to simulated environments and real-world interactive contexts.

Tongyi DeepResearch establishes a new state-of-the-art with substantially fewer parameters, comprising a total of 30.5 billion parameters while activating only 3.3 billion per token, building upon the Qwen3-30B-A3B-Base model (Yang et al., 2025). Empirical evaluations on deep research benchmarks demonstrate the effectiveness of our agent. Tongyi DeepResearch reaches 32.9 on Humanity's Last Exam, 43.4 on BrowseComp, 46.7 on BrowseComp-ZH, 72.2 on WebWalkerQA, 70.9 on GAIA, 75.0 on xbench-DeepSearch, 90.6 on FRAMES and 55.0 on xbench-DeepSearch-2510, outperforming strong baselines such as OpenAI-o3 (OpenAI, 2025b) and Deepseek-V3.1 (DeepSeek Team, 2025). We also provide a systematic analysis covering agentic reinforcement learning, synthetic data, offering key insights into the development of deep research agent. In addition, we present the performance of Tongyi DeepResearch on general benchmarks, including AIME25, HMMT25 and SimpleQA. We believe that agentic models represent an emerging trend for the future, as models increasingly internalize agent-like capabilities and can autonomously invoke the appropriate tools to solve a wide range of problems.

In the following sections, we first outline the design principles underlying Tongyi DeepResearch. We then describe the training pipeline, followed by a comprehensive evaluation of its performance. We

release the model, framework, and end-to-end solutions to support and accelerate community research. This technical report summarizes our main insights and aims to inspire further progress toward scalable and capable agentic systems.

2 Design Principle

Agent Training Pipeline. Agent training is inherently more complex and challenging than conventional LLM training. We introduce two stages in our agent training pipeline: mid-training and post-training. We integrate mid-training directly into the deep research training process, and co-design the end-to-end on-policy reinforcement learning algorithm and its underlying infrastructure for seamless scalability and stability. While most work only applies post-training phase for DeepResearch agents, we novelly introduce mid-training for agentic learning. General foundation models usually lack agentic inductive bias. Most general foundation models are typically pretrained on plain text crawled from the internet and then post-trained on instruction-following data. These datasets lack research-level questions and agentic behaviors, resulting in the model learns agentic capabilities and alignment simultaneously during the post-training phase. Agentic post-training on these general foundation models can result in sub-optimal outcomes and inherent optimization conflicts. Mid-training endows the pre-trained base model with substantial agentic prior knowledge, thereby bridging the gap between pretraining and agentic post-training. Mid-training phase provides a powerful **agentic foundation model** to support effective agentic post-training. During post-training, the model further internalizes deep research capabilities through reinforcement learning with supervised fine-tuning (SFT) for cold start. SFT teaches the model to reliably imitate curated demonstrations, establishing a stable behavioral baseline for research workflows and tool use. However, behavior cloning alone tends to produce mimicry without exploration. RL closes the loop with the environment, using reward signals to refine policies and to internalize agentic planning and execution. In particular, reinforcement learning (1) explores optimal strategies through active interaction with the environment; (2) internalizes goal-directed planning and execution capabilities; and 3) achieves superior sample efficiency by prioritizing high-reward behaviors. The agent first acquires general agentic pattern during supervised fine-tuning phase, while reinforcement learning phase effectively pushes the limits of its agentic performance.

Synthetic Data Centric Scaling. Data serves as the foundation of training, while collecting data for DeepResearch problems is extremely hard. Deep research problems require agents' capability of connecting information, reasoning across sources and validating conclusions. Unlike pre-training data, which is naturally abundant, and conventional LLM post-training data, which is relatively easy to annotate, agentic data is inherently scarce. Research-level problems are difficult to obtain through natural texts from the web. Manually annotating these problems and agentic trajectories is extremely time-consuming and costly (Wei et al., 2025). Building on the aforementioned agent training pipeline, agentic mid-training requires large-scale, diverse trajectories to align subsequent agent behaviors, while agentic post-training depends on high-quality, verifiable data to provide reliable reward signals. As a result, it is hard to rely on natural data to scale DeepResearch capability. Therefore, we focus on synthetic data with large language models. Synthetic data contains several advantages over human annotations below:

- **Synthesizing research-level questions is easy to scale.** We can use LLMs to synthesize question-answer pair efficiently compared to manually annotating.
- **The pattern and diversity are easy to generalize.** LLMs are easy to understand the structure of hard problems and usually have rare insight into diverse patterns, while training annotators to understand the structure and patterns for research-level problems is time-consuming.
- **Synthesized data enables targeted meta-capability enhancement.** By decomposing complex agent tasks into fundamental meta-capabilities (*e.g.*, planning, information synthesis, memory management), we can generate synthetic data that specifically targets and strengthens individual agent skills.
- **Synthesized data can be verified easily.** It is much easier than finding the solution to the question,

which is essential in human annotating.

- **Synthesized data can provide data flywheels in training stages.** After one round of the agentic training pipeline, the trained agentic model can generate synthesized data with stronger reasoning and planning patterns. Data flywheel makes the agentic model evolve iteratively.

Based on these insights, we believe synthetic agentic data becomes the key to scaling deep-research agents. The synthetic data in all phases of the agentic training pipeline are designed in three steps: (1) synthesizing research-level questions; (2) Generating agentic behavior data; (3) Utilizing agentic data in training pipeline.

Learning Through Environmental Interaction. Environmental interaction plays a crucial role in agent intelligence emergence (Silver & Sutton, 2025). However, relying solely on real-world environments for the whole agent training stage faces fundamental challenges: **(1) Non-stationarity**. The dynamic nature of environments causes continuous distribution shift in training data, undermining learning stability; **(2) Interaction cost**. The tangible expense of each API call makes large-scale exploration economically prohibitive. These barriers render agent capability acquisition from the real world alone a formidable endeavor.

In Tongyi DeepResearch, we propose a fundamental reframing: *environments should not be passively viewed as external reality, but actively designed as systems deeply coupled with the training process*. Specifically, we model environments into three forms, each striking a distinct balance between stability, fidelity, and cost:

- **Prior World Environment.** This environment provides task elements, tools, and state definitions, allowing agents to autonomously mine interaction trajectories based on pretrained knowledge without receiving actual environmental responses. It offers perfect stability, zero interaction cost, and unlimited scalability, but lacks real-world feedback signals.
- **Simulated Environment.** This environment constructs controlled, reproducible replicas of real-world interactions locally. It provides stability, rapid response, and low cost, enabling fast iteration and causal attribution analysis. However, its data coverage is inherently limited, exhibiting a notable sim-to-real gap.
- **Real-world Environment.** This environment delivers the most authentic data distribution and feedback signals, serving as the ultimate proving ground for agent capabilities. Its advantage lies in absolute distributional fidelity; the cost is expensive interactions, significant non-stationarity, and exploration risks.

Building on this environmental insight, we adopt adaptive strategies for synthetic data generation and training. Specifically, (1) During agentic mid-training, we primarily leverage the Prior World Environment and Simulated Environment to generate large-scale synthetic data at minimal cost, ensuring efficient agentic ability bootstrapping; (2) During agentic post-training, we validate training strategies and algorithmic techniques in the simulated environment, then deploy verified optimal policies to the real environment for final training. The choice of environments plays a crucial role, agentic intelligence emerges not from a single wolrd, but from carefully chosen environments.

Agent training fundamentally depends on synthetic data and environment interaction. Based on these design principles, we then introduce Tongyi DeepResearch in detail below.

3 Tongyi DeepResearch

3.1 Formulation

We formally define the Tongyi DeepResearch's rollout at each timestep t through three fundamental components:

- **Thought (τ_t)**: The internal cognitive process of the agent. This includes analyzing the current context,

recalling information from memory, planning subsequent steps, and engaging in self-reflection to adjust its strategy.

- **Action (a_t)**: An external operation executed by the agent to interact with its environment. Tongyi DeepResearch is equipped with a versatile set of tools that define its action space, enabling it to interact with a wide range of information sources: *Search*, *Visit*, *Python Interpreter*, *Google Scholar* and *File Parser*. Actions encompass all intermediate tool calls and the final response to the user. In a given trajectory, intermediate actions (a_t where $t < T$) are tool calls, while the final action, a_T, constitutes the generation of an in-depth report for the user.
- **Observation (o_t)**: The feedback received from the environment after an action is performed. This new information is used to update the agent's internal state and inform its next thought.

Based on the fundamental components above, we define two different rollout types as follows:

ReAct. Tongyi DeepResearch's architecture is fundamentally based on the vanilla ReAct (Yao et al., 2023) framework, which synergizes reasoning and acting. In this paradigm, the agent generates both a reasoning trace (Thought) and a subsequent Action in an interleaved manner. This process forms a trajectory, $\mathcal{H}_T$, which is a sequence of thought-action-observation triplets:

$$\mathcal{H}_T = (\tau_0, a_0, o_0, \ldots, \tau_i, a_i, o_i, \ldots, \tau_T, a_T), \tag{1}$$

where a_T represents the final answer to the given task. At any given step $t \leq T$, the agent's policy, π, generates the current thought τ_t and action a_t based on the history of all previous interactions, $\mathcal{H}_{t-1}$:

$$\tau_t, a_t \sim \pi(\cdot|\mathcal{H}_{t-1}). \tag{2}$$

While more complex single and multi-agent paradigms have emerged, our choice of ReAct is a deliberate one, rooted in its simplicity and alignment with fundamental principles. This decision is informed by "The Bitter Lesson" (Sutton, 2019), which posits that general methods leveraging scalable computation ultimately outperform approaches that rely on complex, human-engineered knowledge and intricate designs. Frameworks that require extensive, specialized prompt engineering or possess rigid operational structures risk becoming obsolete as the intrinsic capabilities of models scale (Li et al., 2025a).

Context Management. The execution of long-horizon tasks is fundamentally constrained by the finite length of the agent's context window. To mitigate the risk of context overflow and ensure task focus, we propose the context management paradigm (Qiao et al., 2025), which employs a dynamic context management mechanism based on Markovian state reconstruction. Within this framework, the agent is not conditioned on the complete history. Instead, at each step t, it is conditioned on a strategically reconstructed workspace containing only essential elements: the question q, an evolving report S_t serving as compressed memory, and the immediate context from the last interaction (a_t and o_t). This Markovian structure enables the agent to maintain consistent reasoning capacity across arbitrary exploration depths while naturally circumventing the degradation. For every step $0 < t < T$, this core update process can be formalized as:

$$S_t, \tau_{t+1}, a_{t+1} \sim \pi(\cdot|S_{t-1}, a_t, o_t). \tag{3}$$

This context management paradigm is particularly crucial, it not only prevents context suffocation but also enforces structured reasoning by requiring the agent to explicitly synthesize and prioritize information at each step. This design naturally aligns with human research patterns, where periodic synthesis and reflection are essential for maintaining coherent long-term investigation.

3.2 Overall Training Recipe

The system is initialized from the pretrained base model Qwen3-30B-A3B-Base[1]. Tongyi DeepResearch is developed through an end-to-end training framework that integrates agentic mid-training and post-training, enabling scalable reasoning and information seeking across complex research tasks. This

[1] https://huggingface.co/Qwen/Qwen3-30B-A3B-Base

establishes a new paradigm for training agentic models. We first present the mid-training process in Section 3.3, followed by the post-training stage in Section 3.4.

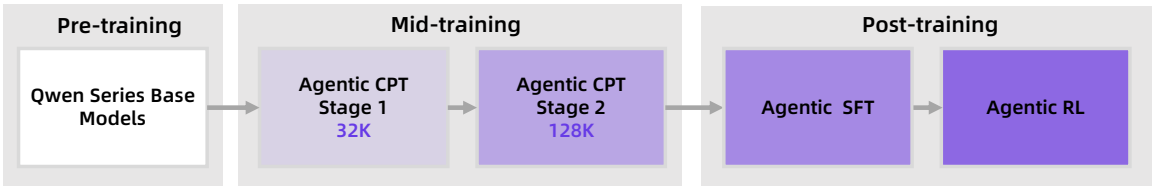


Figure 2: Training pipeline of Tongyi DeepResearch.

3.3 Agentic Mid-training

3.3.1 Training Configuration

Tongyi DeepResearch employs a two-stage **Agentic Continual Pre-training (Agentic CPT)** (Su et al., 2025) as its core *mid-training* phase. This phase functions as a critical bridge connecting pre-trained models and agentic post-training. Its primary objective is to provide a base model endowed with a strong inductive bias for agentic behavior, while simultaneously preserving broad linguistic competence. To achieve this, the optimization process is driven by the standard *Next-Token Prediction* loss function.

The design of this phase is strategically optimized for both efficiency and progressive capability scaling. We initiate with a **32K** context length in the first stage, before expanding to **128K** in the second. This expanded context window is specifically leveraged in the second stage, where we introduce a substantial corpus of long-sequence (64K-128K) agentic behavior data. This approach is critical for enhancing the model's capacity for coherent, long-horizon reasoning and action. Throughout both stages, a small proportion of general pre-training data is interleaved, ensuring the model acquires specialized agentic competence without sacrificing its foundational generalization capabilities.

3.3.2 Large-scale Agent Behavior Data Synthesis

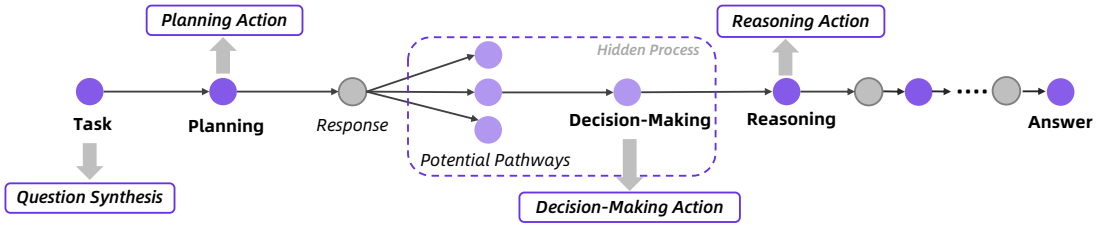


Figure 3: Large-scale agent behavior data synthesis for agentic continual pre-training.

In Agentic CPT, we synthesize data across the complete lifecycle of agent workflows as shown in Figure 3. A typical agent workflow begins with a problem, iteratively cycles through reflection and action, and ultimately converges on a final solution. To comprehensively capture this process, we synthesize data for the critical steps that constitute the agent's operational cycle: Question Synthesis, Planning Action, Reasoning Action, and Decision-Making Action. Note that while decision-making is often implicit within agent cycles, we explicitly model it as a distinct action type in our synthesis framework.

Large-scale Multi-style Question Synthesis. Grounded in continuously updated open-world knowledge, we construct an entity-anchored open-world memory. This memory consolidates diverse real-world knowledge sources, such as web-crawled data and agent interaction trajectories, into structured representations of entities and their associated knowledge. Building upon this foundation, we sample entities along with their related knowledge to generate diverse questions that embed specific behavioral pattern requirements, such as multi-hop reasoning questions and numerical computation questions.

Planning Action. Planning refers to problem decomposition and first-step action prediction. A key insight is that planning accuracy is highly correlated with whether an agent can successfully complete a task. Thus, we employ open-source models to analyze, decompose, and predict initial actions for the synthesized questions. Furthermore, we leverage the entities and associated knowledge used in question construction as the basis for rejection sampling, thereby ensuring high-quality planning outputs.

Reasoning Action. Logical reasoning and knowledge integration over heterogeneous data is foundational for agents solving complex tasks. When external tools return massive unstructured responses, whether models can distill critical knowledge from noise and construct coherent reasoning paths directly determines task outcomes. To this end, given a question and its dependent knowledge, we guide large models through a two-stage process to generate complete reasoning chains, with a dual filtering mechanism based on reasoning length and answer consistency to ensure quality.

Decision-Making Action. Each step of an agent's thinking and action is essentially an implicit decision-making process. Specifically, each decision point encompasses multiple potential reasoning and action paths, from which the agent must select the most promising solution. To capture this critical mechanism, we explicitly model this decision-making process. First, based on existing demonstration trajectories, we thoroughly explore the feasible action space at each step. Second, we reconstruct the original trajectories into multi-step decision sequences while preserving the original decision choices.

General Function-calling Data Synthesis via Environment Scaling. To enhance our model's general agentic capability, we systematically scale the function-calling data through environment scaling. The breadth of function-calling competence is closely tied to the diversity of environments in which agents are trained (Fang et al., 2025). We also scale up environments as a step towards advancing general agentic intelligence. In designing environment construction and scaling, we follow the principle that the core of an agent lies in its capacity for environment interaction, with each environment instantiated as a *read–write* database. We design a scalable framework that automatically constructs heterogeneous environments that are fully simulated, systematically broadening the space of function-calling scenarios. The produced data are incorporated into the model's mid-training phase.

3.4 Agentic Post-training

The post-training pipeline comprises three stages: data synthesis, supervised fine-tuning for cold start, and agentic reinforcement learning.

3.4.1 High-quality Data Synthesis

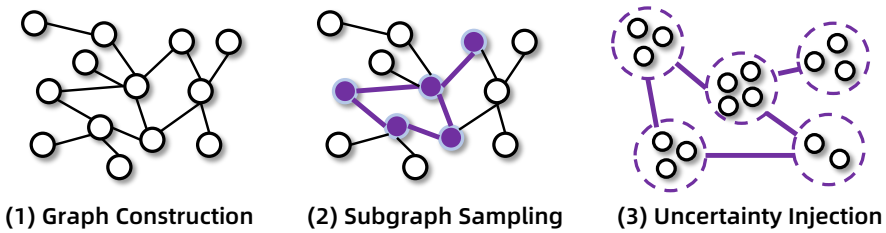


Figure 4: High-quality data synthesis pipeline.

We develop an end-to-end solution for synthetic data generation to generate complex, high-uncertainty and super-human level question and answer pairs (Li et al., 2025c;b), as shown in Figure 4. This fully automated process requires no human intervention to construct super-human quality datasets, designed to push the boundaries of agent performance. The process begins by constructing a highly interconnected knowledge graph via random walks, leveraging web search to acquire relevant knowledge, and isomorphic tables from real-world websites, ensuring a realistic information structure. We then

sample subgraphs and subtables to generate initial questions and answers. The pivotal step involves strategically increasing the uncertainty within the question to enhance its difficulty (Wu et al., 2025a). This practical approach is grounded in a complete theoretical framework, where we formally model QA difficulty as a series of controllable "atomic operations" (*e.g.*, merging entities with similar attributes) on entity relationships, allowing us to systematically increase complexity. To further reduce inconsistencies between the organized information structure and the reasoning structure of QA, enable more controllable difficulty and structure scaling of reasoning, we proposed a formal modeling of the information-seeking problem based on set theory (Tao et al., 2025). With this formalization, we develop agents that expands the problem in a controlled manner, and minimizes reasoning shortcuts and structural redundancy, leading to further improved QA quality. Moreover, this formal modeling also allows for efficient verification of QA correctness, effectively addressing the challenge of validating synthetic information-seeking data for post-training.

We also develop an automated data engine to scale the generation of PhD-level research questions (Qiao et al., 2025). Starting from a multi-disciplinary knowledge base, it creates seed QA pairs requiring multi-source reasoning. These seeds undergo iterative complexity upgrades, where a question-crafting agent, equipped with the corresponding tool, progressively expands scope and abstraction. Each iteration refines and compounds prior outputs, enabling a systematic and controllable escalation of task difficulty.

3.4.2 Supervised Fine-tuning for Cold Start

The initial phase of our agentic post-training pipeline is a supervised fine-tuning (SFT) stage, designed to equip the base model with a robust initial policy prior to reinforcement learning. Starting from our synthesized high-quality QA data, we obtain training trajectories that cover the complete thought process and tool responses generated by high-performing open-source models, which are then subjected to a rigorous rejection sampling protocol. This comprehensive filtering process guarantees that only high-quality trajectories exhibiting diverse problem-solving patterns are retained.

Mixed Training Paradigm. The cold stage training leverages data from two different formulations to enhance model robustness and generalization. For the React Mode, the training samples take the historical state $\mathcal{H}_{t_1}$ as input, and output the corresponding thought τ_i and tool call a_i for the current step. For our Context Management Mode , the training samples take as input the previous step's trajectory summary S_{t-1}, tool call a_{i-1}, and tool response o_{i-1}, and output the current step's trajectory summary, thought τ_i, and tool call a_i. The Context Management Mode data particularly strengthens the agent's capabilities in state analysis and strategic decision-making, as it requires the model to synthesize complex observations into coherent summaries while maintaining task focus across extended trajectories. This synthesis-oriented training enables more deliberate reasoning patterns compared to purely ReAct. We adopt a two-stage training strategy based on context length. In the first stage, the context length is set to **40K**, and the training data consist of ReAct Mode samples with context lengths shorter than 40K, along with all Context Management Mode samples (as they are all within 40k). In the second stage, the context length is extended to **128K**, and the training data include ReAct Mode samples with context lengths between 40K and 128K, as well as a small portion of 40K data for stability.

3.4.3 Agentic Reinforcement Learning

To advance the model's capabilities toward more robust and reliable planning and searching in a complex web environment, we apply an agentic RL framework, which is illustrated in Figure 5. In this framework, the model generates a complete task attempt (a "rollout") and receives a reward if its final answer matches the ground truth (RLVR) (Guo et al., 2025). Throughout this agentic RL procedure, the model continuously interacts with the environment (simulated or real-world), iteratively refining its policy with each iteration, and, in turn, using that improved policy to curate a new, higher-quality set of training data.

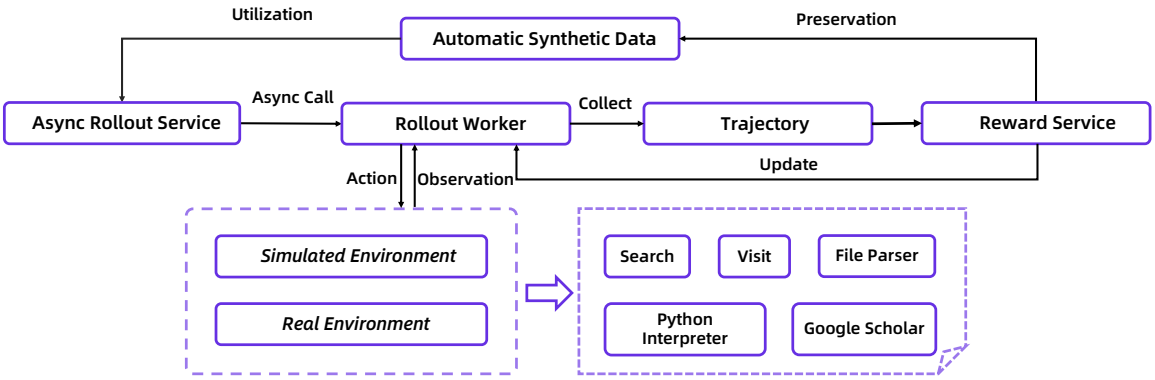


Figure 5: An overview of our agentic reinforcement learning framework.

Real-world Environment. Our agent's toolkit is a complex system that integrates several specialized tools[2]: (1) **Search**, (2) **Visit**, (3) **Python Interpreter**, (4) **Google Scholar**, (5) **File Parser**. The end-to-end reliability of this system is paramount. The inherent volatility of external APIs, encompassing high latency, outright failures, and inconsistent returns, threatens to corrupt our training trajectories. This data contamination makes it nearly impossible to diagnose performance issues, obscuring whether a poor outcome is caused by a weakness in the agent's policy or by the instability of the environment itself. To ensure reliable tool use during agent training and evaluation, we developed a unified sandbox. This interface is built around a central scheduling and management layer that orchestrates every tool call. For each tool, we implement robust concurrency controls and fault-tolerance mechanisms, such as proactive QPS rate constraints, result caching, automatic timeout-and-retry protocols, graceful service degradation for non-critical failures, and seamless failover to backup data sources (*e.g.*, a backup search API). This design abstracts the tool invocation into a deterministic and stable interface for the agent and thereby insulates the training loop from real-world stochasticity while also significantly reducing operational costs. This design abstracts tool invocation into a deterministic interface, providing a stable and fast experience that is crucial for preventing tool errors from corrupting the agent's learning trajectory.

Simulated Environment. Directly utilizing real-world web environment APIs presents numerous practical problems[3]. We first build an offline environment based on the 2024 Wikipedia database and develop a suite of local RAG tools to simulate the web environment. We then reuse the data synthesis pipeline to create a high-quality, structurally complex QA specifically for this offline environment. This provides us with a low-cost, high-efficiency, and fully controllable platform that enables high-frequency, rapid experimentation, thereby greatly accelerating our development and iteration process.

On-Policy Asynchronous Rollout Framework. The iterative nature of agentic rollouts, which require numerous interactions with the environment, creates a significant bottleneck that slows down the entire RL training process. To overcome this, we implement a custom, step-level asynchronous RL training loop built on the rLLM framework (Tan et al., 2025). Our solution utilizes two separate asynchronous online servers, with one for model inference and another for tool invocation. A centralized interaction handler then processes the outputs from both, formatting the feedback into a unified message list. This architecture allows multiple agent instances to interact with the environment in parallel, each completing its rollout independently.

[2]The details for each tool are shown in Appendix D.

[3]Queries per second (QPS) impact significantly degrade our development efficiency and compromise the reliability during our early-stage ablation studies.

RL Training Algorithm. Our RL algorithm is a tailored adaptation of GRPO (Shao et al., 2024):

$$\mathcal{J}(\theta) = \mathbb{E}_{(q,y)\sim\mathcal{D},\{\mathcal{H}^i\}_{i=1}^{G}\sim\pi_{\theta_{\text{old}}}(\cdot|context)} \left[\frac{1}{\sum_{i=1}^{G}|\mathcal{H}^i|} \sum_{i=1}^{G}\sum_{j=1}^{|\mathcal{H}^i|} \min\left(r_{i,j}(\theta)\hat{A}_{i,j},\ \text{clip}\Big(r_{i,j}(\theta), 1-\varepsilon_{low}, 1+\varepsilon_{high}\Big)\hat{A}_{i,j}\right) \right], \tag{4}$$

where (q, y) is the question-answer pair, $r_{i,j}(\theta)$ is the importance sampling ratio (remains 1.0 for strictly on-policy training), and $\hat{A}_{i,j}$ is an estimator of the advantage at token j:

$$r_{i,j}(\theta) = \frac{\pi_\theta(\mathcal{H}^{i,j} \mid context)}{\pi_{\theta_{\text{old}}}(\mathcal{H}^{i,j} \mid context)}, \quad \hat{A}_{i,j} = R_i - \text{mean}(\{R_i\}_{i=1}^{G}). \tag{5}$$

We employ a strict on-policy regimen, where trajectories are consistently sampled using the most up-to-date policy, ensuring that the learning signal is always relevant to the model's current capabilities. The reward is a pure 0 or 1 signal of answer correctness. We **do not** include a format reward (*e.g.*, 0.1 for format correctness) because the preceding cold start stage ensures the model is already familiar with the required output format. Following DAPO (Yu et al., 2025), we apply the token-level policy gradient loss in the training objective and clip-higher strategy to encourage more exploration. To further reduce variance in the advantage estimation, we adopt a leave-one-out strategy (Chen et al., 2025). Furthermore, we observed in preliminary experiments that directly optimizing on an unfiltered set of negative rollouts significantly degrade training stability and can lead to policy collapse after extended training. To mitigate this, we selectively exclude certain negative samples from the loss calculation, for instance, those that do not yield a final answer because they exceed a length limit. The primary motivation for these modifications is not algorithmic novelty but the pragmatic pursuit of a more efficient and stable training paradigm.

Automatic Data Curation. We optimize data in real time, guided by training dynamics to generalize to out-of-distribution scenarios through self-exploration. This optimization is achieved through a fully automated data filtering pipeline that dynamically adjusts the training set based on the improved policy model. Specifically, our process begins with a large dataset, $\mathcal{D}$. We use the initial SFT model as a baseline policy to sample multiple solution attempts, or rollouts, for each problem. We then create an initial training set, $\mathcal{D}'$, by filtering out problems where the model either always fails or always succeeds, as these will offer no learning signal for RL training. This leaves us with a focused subset of problems of moderate difficulty. During RL training, we continuously monitor the problems in $\mathcal{D}'$ by their latest rollouts to see if they have become too easy for the improved policy model. In parallel, a separate process uses intermediate checkpoints of the policy model to sample from the entire original dataset, $\mathcal{D}$. This background process identifies and collects a backup pool of new problems that have become moderately difficult for the now-stronger model. When the training reaches a certain step count or the reward plateaus, we refresh the active training set $\mathcal{D}'$ by removing the mastered problems and incorporating new, challenging ones from the backup pool. The entire data filtering and refreshment pipeline runs independently, never interrupting the main RL training loop. This design allows us to automatically evolve both the policy model and its training data, ensuring consistently high training efficiency and stability.

Through our experiments, we arrive at a critical insight: **the success of agentic RL depends more on the quality of the data and the stability of the training environment than on the specific algorithm being used**. Consequently, we concentrate our efforts on designing a stable environment and curating high-quality data, making only a few essential modifications to the algorithm itself, mainly for the purpose of stabilizing the training process.

3.4.4 Model Merging

We employ model merging at the last stage of the pipeline. This approach is built on the key insight that when different model variants are derived from the same pre-trained model, their parameters can be effectively combined through averaging or interpolation (Wang et al., 2025). Specifically, our process involves selecting several model variants that originate from the same base model but exhibit different capability preferences. We then create the final merged model by computing a weighted average of their parameters:

$$\theta_{\text{merged}} = \sum_k \alpha_k \cdot \theta^{(k)}, \quad \text{s.t.} \quad \sum_k \alpha_k = 1,\ \alpha_k \geq 0. \tag{6}$$

where $\theta^{(k)}$ represents the parameters of the k-th model variant, and α_k is its corresponding merge weight. Empirically, this interpolation strategy not only preserves the core strengths of each contributing model but also equips the merged model with robust generalization abilities. In complex scenarios requiring a synthesis of these varied capabilities, the merged model performs comparably to the best-performing source model in its respective area of strength, all without incurring additional optimization costs.

4 Experiments

4.1 Experimental Setup

Backbones. We evaluate Tongyi DeepResearch on seven public information-seeking benchmarks spanning long-term reasoning and long-horizon tool use. The model is compared against two families of systems: 1) LLM-based ReAct agents: GLM-4.5 (Zeng et al., 2025), Kimi-K2 (Team et al., 2025), DeepSeek-V3.1 (DeepSeek Team, 2025), Claude-4-Sonnet (anthropic, 2025), OpenAI o3/o4-mini (OpenAI, 2025b)) and 2) end-to-end deep-research agents: OpenAI DeepResearch (OpenAI, 2025a), Gemini DeepResearch (Gemini Team, 2025), Kimi Researcher (Kimi, 2025).

Benchmarks. We follow each benchmark's official evaluation protocol. The benchmarks cover: (1) Humanity's Last Exam (Phan et al., 2025); (2) BrowseComp (Wei et al., 2025) and BrowseComp-ZH (Zhou et al., 2025); (3) GAIA (Mialon et al., 2023); (4) xBench-DeepSearch (Xbench Team, 2025); (5) WebWalkerQA (Wu et al., 2025b); (6) FRAMES (Krishna et al., 2025); and (7) xbench-DeepSearch-2510.

All scores are computed with the official scripts released by each benchmark. The details of evaluation are presented in Appendix B.

Evaluation. We adopt fixed inference parameters to ensure stability and reproducibility across evaluations: temperature = 0.85, repetition penalty = 1.1, and top-p = 0.95. A maximum of 128 tool invocations is allowed per task, and the context length is constrained to 128K tokens. Each benchmark is evaluated three times independently, and we report the average performance (`Avg@3`) as the main metric. For completeness, we also report the best `Pass@1` (best result over 3 runs) and `Pass@3` results in the subsequent analysis. All results are obtained on September 16, 2025, except for xbench-DeepSearch-2510, which is evaluated on October 28, 2025.

Reproduce. Tongyi DeepResearch operates utilizing an action space that includes the Search, Visit, Python, Scholar, and File Parser tools. We release official reproduction scripts on GitHub[4], along with the complete tool implementations and prompt configurations.

4.2 Main Results

Table 1 presents the performance of Tongyi DeepResearch compared with a broad range of state-of-the-art LLM-based agents and proprietary deep research systems across multiple benchmarks, including Humanity's Last Exam, BrowseComp, BrowseComp-ZH, GAIA, xbench DeepSearch, WebWalker QA, and FRAMES. Tongyi DeepResearch achieves the highest scores on nearly all evaluated benchmarks,

[4] https://github.com/Alibaba-NLP/DeepResearch

Table 1: Performance comparison on various benchmarks.

Benchmarks	Humanity's Last Exam	Browse Comp	Browse Comp-ZH	GAIA	xbench DeepSearch	WebWalker QA	FRAMES
LLM-based ReAct Agent							
GLM 4.5	21.2	26.4	37.5	66.0	70.0	65.6	78.9
Kimi K2	18.1	14.1	28.8	57.7	50.0	63.0	72.0
DeepSeek-V3.1	29.8	30.0	49.2	63.1	71.0	61.2	83.7
Claude-4-Sonnet	20.3	12.2	29.1	68.3	65.0	61.7	80.7
OpenAI o3	24.9	49.7	58.1	–	67.0	71.7	84.0
OpenAI o4-mini	17.7	28.3	–	60.0	–	–	–
DeepResearch Agent							
OpenAI DeepResearch	26.6	51.5	42.9	67.4	–	–	–
Gemini DeepResearch	26.9	–	–	–	–	–	–
Kimi Researcher	26.9	–	–	–	69.0	–	78.8
Tongyi DeepResearch (30B-A3B)	**32.9**	**43.4**	**46.7**	**70.9**	**75.0**	**72.2**	**90.6**

demonstrating strong generalization across both English and Chinese tasks. It consistently surpasses both open and closed commercial systems, including OpenAI o3, DeepSeek-V3.1, and Gemini DeepResearch. On the newly released xbench-DeepSearch-2510, Tongyi DeepResearch ranks just below ChatGPT-5-Pro, demonstrating competitive performance at the forefront of the field. Notably, these gains are achieved with only 3.3 billion activated parameters per token, underscoring the model's efficiency and scalability. In aggregate, Tongyi DeepResearch sets a new state of the art among open-source deep research agents, narrowing and in some cases even surpassing the performance of frontier proprietary systems while maintaining superior interpretability and computational efficiency.

4.3 Heavy Mode

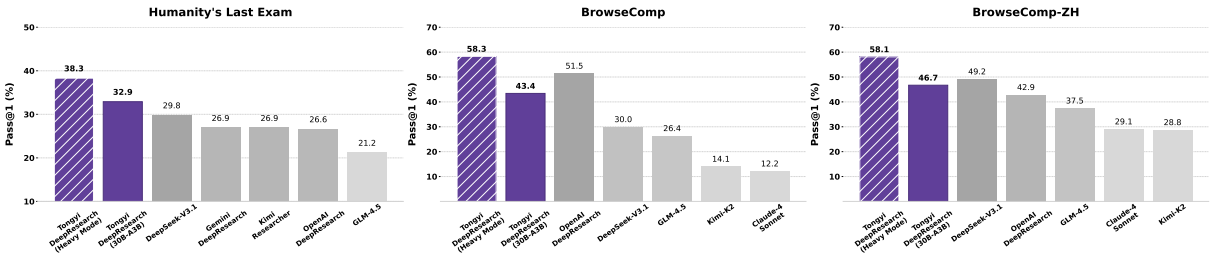


Figure 6: Performance comparison between Tongyi DeepResearch Heavy Mode and state-of-the-art models.

To further unlock the potential of deep research agents, we introduce the **Heavy Mode**, which leverages test-time scaling through a Research-Synthesis framework built upon the context management paradigm. Given that DeepResearch involves multi-round tool calls and intensive reasoning, directly aggregating contexts from multiple trajectories is computationally prohibitive. Our Heavy Mode addresses this challenge through strategic parallelization and synthesis.

Parallel Research Phase. We deploy n parallel agents, each following the context management paradigm but exploring diverse solution paths through different tool usage and reasoning strategies. Each agent u independently processes the question q and produces a final report and answer:

$$(S_T^u, \text{answer}_u) = \text{Agent}_u(q), \quad u \in [1, n] \tag{7}$$

where S_T^u represents the final report summary from agent u after T iterations, encapsulating the complete reasoning trajectory in compressed form.

Integrative Synthesis Phase. A synthesis model consolidates all parallel findings to produce the final answer:

$$\text{answer}_{\text{final}} = \text{Synthesis}\left(\{(S_T^u, \text{answer}_u)\}_{u=1}^{n}\right), \tag{8}$$

The key advantage of this approach lies in the compressed nature of context management reports S_T^u. Unlike traditional methods that would require aggregating full trajectories (potentially exceeding context limits with just 2-3 agents), our approach enables the synthesis model to assess n diverse solution strategies within a manageable context window. Each report S_T^u preserves the essential reasoning logic and findings while discarding redundant intermediate steps, enabling effective test-time scaling.

As shown in Figure 6, our Heavy Mode achieves state-of-the-art performance on Humanity's Last Exam (38.3%) and BrowseComp-ZH (58.1%), while remaining highly competitive on BrowseComp (58.3%). These substantial improvements validate the effectiveness of our heavy mode based on context management in leveraging test-time compute through parallel exploration and intelligent aggregation.

4.4 Detailed Analysis

Pass@1 and Pass@3 Performance. We report the `Avg@3` performance in Table 1. Given the dynamic and complex nature of agent environments, we further conduct a fine-grained analysis of `Pass@1` (over three runs) and `Pass@3` in Figure 7. Despite the unstable evaluation environment, our final `Avg@3` results are consistent with the `Pass@1` (best result over 3 runs) results, demonstrating the robustness of our deep research approach. Our `Pass@3` performance demonstrates the strong potential of our agent. In particular, it achieves **59.64 on BrowseComp**, **63.67 on BrowseComp-ZH**, and **45.9 on Humanity's Last Exam**.

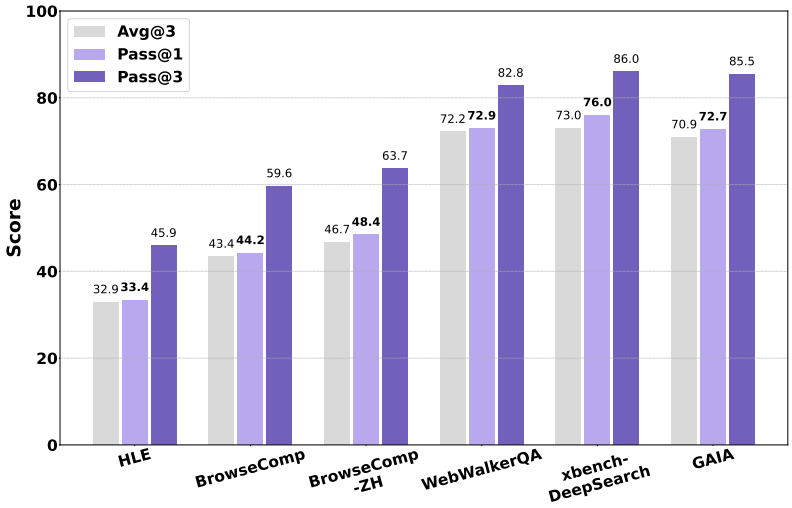


Figure 7: Detailed evaluation results using `Avg@3`, `Pass@1` and `Pass@3` metric.

Training Rewards and Entropy. As shown in Figure 8, the agent's performance exhibits a clear and significant upward trend with training, confirming effective policy learning. The sustained nature of this improvement underscores the success of our dynamic data curation, which prevents learning from stagnating by consistently providing challenging material. Concurrently, the policy entropy exhibits exceptional stability, converging to a consistent value after a brief initial increase and thereby avoiding both collapse and explosion. This outcome serves as strong evidence for our methodological contributions in environment design and algorithm modification, which together create the necessary conditions for a remarkably stable and effective RL training paradigm.

Context Length of RL. In Figure 10, we analyze the impact of the model's context length on the agentic RL training process, comparing models with 32k, 48k, and 64k context limits. It is important to note

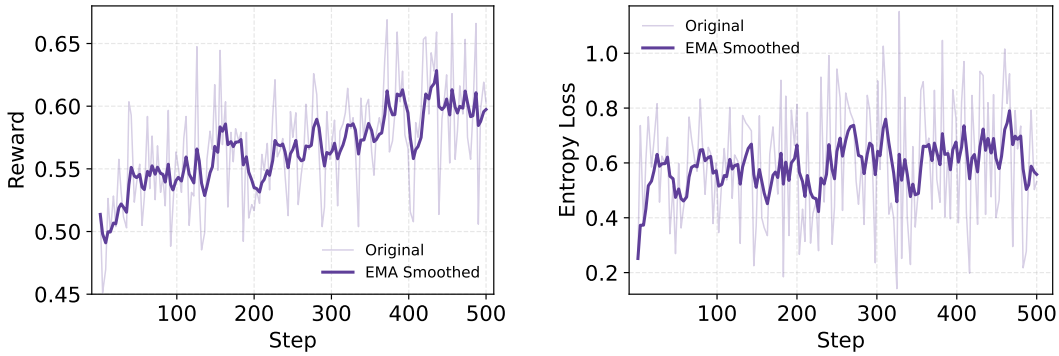


Figure 8: Reward and entropy loss of agentic RL training.

that the dynamic data curation for all three experimental variants was performed using the same model with a 64k context. Focusing first on the reward dynamics in the left panel, we observe that all three models demonstrate effective and stable policy learning, evidenced by a monotonically increasing reward. This confirms the robustness of our training framework. However, their performance ceilings diverge significantly, which is an expected consequence of our data curation method. Because the curriculum is populated with problems deemed moderately difficult by the highly capable 64k context model, many of these problems inherently require long and complex reasoning to solve. Consequently, a clear hierarchy emerges: the 64k model, perfectly matched to its own data, achieves the highest reward. The 48k and 32k models, being increasingly constrained, are unable to solve the most complex problems in the curriculum, thus capping their maximum potential reward.

The training dynamics in the right panel reveal a more interesting story. The model with a 64k context exhibits a steady increase in average response length, learning to leverage its expansive context to build more elaborate solutions. In contrast, the model with a 48k context maintains a consistent equilibrium, improving its policy within a stable complexity budget. Most surprisingly, the model with a 32k context displays a clear downward trend in response length. This observation provides a key insight: for a model with a limited context, RL training on a curriculum designed for a more capable model can force it to discover more efficient solutions. This effect arises because our dynamic data curriculum is continuously updated using the 64k context model, a process that populates the training set with problems whose optimal solutions can be longer than 32k tokens. For the model with a 32k context, attempting these problems is likely to yield a zero-reward signal. This creates a powerful implicit incentive to discover more concise, potent action sequences that fit within its limit, thus becoming more efficient over time.

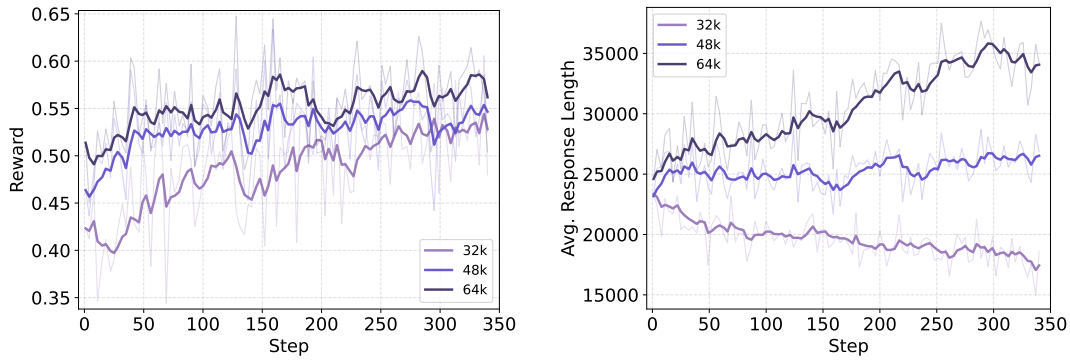
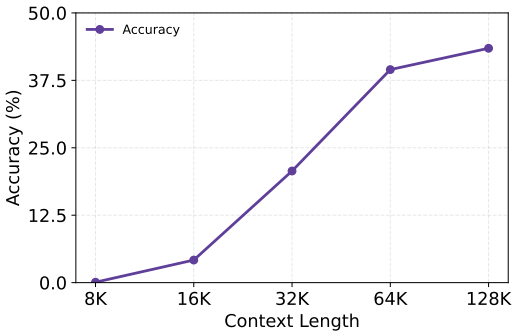
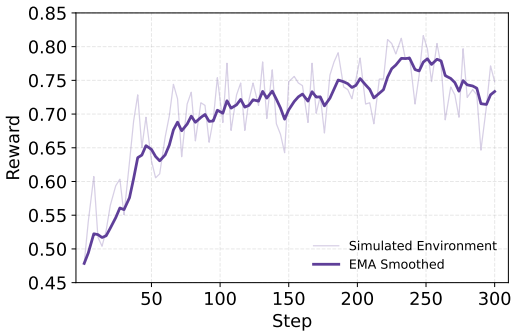


Figure 9: Comparison of different context length limits for RL training.

Interaction Test-time Scaling. Unlike conventional models, the DeepResearch agent primarily relies on interactions with the environment to acquire information and accomplish tasks. Therefore, the number of



(a) Interaction turns scaling for BrowseComp.



(b) Reward in the simulated environment.

Figure 10: Detailed analysis on interaction scaling and simulated environments.

interaction turns with the environment is crucial. While reasoning models can be scaled by increasing the number of output tokens, our approach scales along a different dimension, the number of environment interactions. Naturally, as the number of interactions increases, the agent obtains more observations from environment, resulting in a longer context. Figure 10a illustrates our scaling curve: as the context length and number of interactions grow, the model's performance on the BrowseComp dataset improves consistently.

Super-human Level Synthetic Data. To validate the effectiveness of our synthetic data, we conducted a statistical analysis of the SFT dataset. Over **20%** of the samples exceed 32k tokens and involve more than 10 tool invocations. This demonstrates the high complexity and richness of our synthetic data. Such high-quality, cold-start data provides the model with a strong foundation for deep reasoning and research capabilities, serving as an excellent initialization for the RL phase. During reinforcement learning, we leverage automated data curation to make more effective use of the synthetic data.

From Simulation to Reality. To rapidly validate our algorithm, we built a simulated Wiki environment that mirrors real-world conditions. We test our adapted GRPO algorithm in this environment, and the resulting reward curve, shown in Figure 10b, closely matches the one observed in the real environment, as shown in Figure 8. This Wiki simulation environment provides functionality analogous to a "wind tunnel laboratory", enabling fast algorithm iteration and significantly improved our development efficiency.

Performance on General Benchmark. We evaluate three general benchmarks, AIME25, HMMT25 and SimpleQA (OpenAI, 2025c). The results are shown in Figure 11. Experimental results demonstrate that Tongyi DeepResearch achieves substantial improvements over the base model, which relies solely on reasoning without any tool use. On one hand, the system can retrieve external information via search, which proves particularly effective for knowledge-intensive benchmarks, and on the other, Python Interpreter enables it to enhance performance on mathematical reasoning tasks through native computational support. Looking ahead, model training increasingly converges with agent training, solving paradigms evolve toward agentic architectures that integrate tool invocation and environment interaction, reflecting a more human-like problem-solving process.

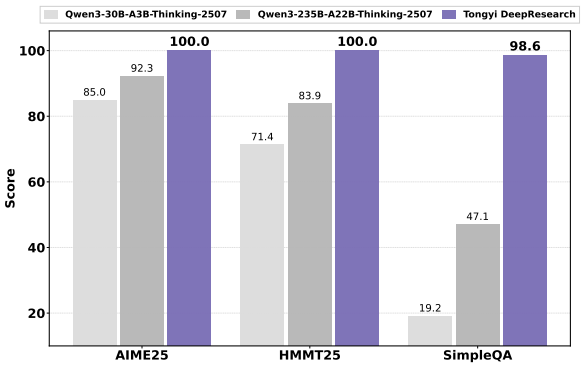


Figure 11: Performance on general benchmarks.

5 Discussion

5.1 Limitations

We acknowledge several limitations in our current work: First, the current 128K context length remains insufficient for handling the most complex long-horizon tasks, motivating further exploration of extended context windows or more advanced context management mechanisms (Qiao et al., 2025; Wu et al., 2025c). Second, we have not yet released a larger-scale model. Although the smaller-sized model already demonstrates strong performance, a larger model is currently in progress. Third, we are continuously improving report generation fidelity and optimizing for user preferences to ensure more faithful, useful, and preference-aligned outputs (Li et al., 2025e). Fourth, we aim to improve the efficiency of our reinforcement learning framework by exploring techniques such as partial rollouts, which will require addressing off-policy training challenges, including distributional shift. Finally, our current Deep Research training focuses on specific prompt instructions and predefined tool sets. We plan to enhance its robustness and extend the framework from Deep Research to broader agentic tool use scenarios.

5.2 Model Scale

We believe that training agentic capabilities on relatively small models is highly valuable (Belcak et al., 2025). Smaller models are inherently more efficient to deploy on edge devices, broaden accessibility across diverse real-world scenarios, and deliver faster, more responsive interactions. This direction aligns with the broader goal of making autonomous research agents both powerful and practically deployable.

5.3 What's Next

We have a long-standing commitment to advancing research and development in deep research agents. The Tongyi DeepResearch represents a significant step toward AI systems capable of autonomously transforming information into insight. We advocate for open-source models with emergent agency, which are essential for democratizing agentic intelligence and deepening our fundamental understanding of how agency can emerge and scale in open systems. Looking ahead, we aim to evolve from domain-specific agents to general-purpose agents, which are capable of reasoning, planning, and acting autonomously across diverse domains with minimal human supervision. To achieve this, we are developing the **next-generation agent foundation model**, a unified model designed to endow AI systems with scalable reasoning, memory, and autonomy, enabling them to operate as truly general agents. We believe it will empower individuals and organizations to reach new heights of productivity and innovation.

6 Conclusion

We introduced Tongyi DeepResearch, an open-source deep research agent that unifies agentic mid-training and post-training into a scalable, end-to-end paradigm. Through automated data synthesis and stage-specific environments, the model learns to plan, search, reason, and synthesize information autonomously. Despite its efficiency, activating only 3.3B parameters, Tongyi DeepResearch achieves state-of-the-art results on multiple deep research benchmarks, surpassing strong proprietary systems. This work establishes a foundation for open, reproducible research into autonomous AI agents and marks a step toward more general, self-improving intelligence.

Contributions

The names are listed in alphabetical order by first name.

Project Leader
Yong Jiang

Core Contributors
Baixuan Li, Bo Zhang, Dingchu Zhang, Fei Huang, Guangyu Li, Guoxin Chen, Huifeng Yin, Jialong Wu, Jingren Zhou, Kuan Li, Liangcai Su, Litu Ou, Liwen Zhang, Pengjun Xie, Rui Ye, Wenbiao Yin, Xinmiao Yu, Xinyu Wang, Xixi Wu, Xuanzhong Chen, Yida Zhao, Zhen Zhang, Zhengwei Tao, Zhongwang Zhang, Zile Qiao

Contributors
Chenxi Wang, Donglei Yu, Gang Fu, Haiyang Shen, Jiayin Yang, Jun Lin, Junkai Zhang, Kui Zeng, Li Yang, Hailong Yin, Maojia Song, Ming Yan, Minpeng Liao, Peng Xia, Qian Xiao, Rui Min, Ruixue Ding, Runnan Fang, Shaowei Chen, Shen Huang, Shihang Wang, Shihao Cai, Weizhou Shen, Xiaobin Wang, Xin Guan, Xinyu Geng, Yingcheng Shi, Yuning Wu, Zhuo Chen, Zijian Li

References

anthropic. Introducing claude 4, 2025. URL https://www.anthropic.com/news/claude-4.

Peter Belcak, Greg Heinrich, Shizhe Diao, Yonggan Fu, Xin Dong, Saurav Muralidharan, Yingyan Celine Lin, and Pavlo Molchanov. Small language models are the future of agentic ai. *arXiv preprint arXiv:2506.02153*, 2025.

Jingyi Chai, Shuo Tang, Rui Ye, Yuwen Du, Xinyu Zhu, Mengcheng Zhou, Yanfeng Wang, Yuzhi Zhang, Linfeng Zhang, Siheng Chen, et al. Scimaster: Towards general-purpose scientific ai agents, part i. x-master as foundation: Can we lead on humanity's last exam? *arXiv preprint arXiv:2507.05241*, 2025.

Kevin Chen, Marco Cusumano-Towner, Brody Huval, Aleksei Petrenko, Jackson Hamburger, Vladlen Koltun, and Philipp Krähenbühl. Reinforcement learning for long-horizon interactive llm agents. *arXiv preprint arXiv:2502.01600*, 2025.

Claude Team. Claude research, 2025. URL https://www.anthropic.com/news/research.

DeepSeek Team. Introducing deepseek-v3.1: our first step toward the agent era!, 2025. URL https://api-docs.deepseek.com/news/news250821.

Runnan Fang, Shihao Cai, Baixuan Li, Jialong Wu, Guangyu Li, Wenbiao Yin, Xinyu Wang, Xiaobin Wang, Liangcai Su, Zhen Zhang, et al. Towards general agentic intelligence via environment scaling. *arXiv preprint arXiv:2509.13311*, 2025.

Gemini Team. Gemini deep research, 2025. URL https://gemini.google/overview/deep-research/.

Grok Team. Grok-3 deeper search, 2025. URL https://x.ai/news/grok-3.

Daya Guo, Dejian Yang, Haowei Zhang, Junxiao Song, Ruoyu Zhang, Runxin Xu, Qihao Zhu, Shirong Ma, Peiyi Wang, Xiao Bi, et al. DeepSeek-R1: Incentivizing reasoning capability in LLMs via reinforcement learning. *arXiv preprint arXiv:2501.12948*, 2025.

Jina.ai. Jina, 2025. URL https://jina.ai/.

Kimi. Kimi-researcher: End-to-end rl training for emerging agentic, 2025. URL https://moonshotai.github.io/Kimi-Researcher/.

Satyapriya Krishna, Kalpesh Krishna, Anhad Mohananey, Steven Schwarcz, Adam Stambler, Shyam Upadhyay, and Manaal Faruqui. Fact, fetch, and reason: A unified evaluation of retrieval-augmented generation. In *Proceedings of the 2025 Conference of the Nations of the Americas Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 1: Long Papers)*, pp. 4745–4759, 2025.

Kuan Li, Liwen Zhang, Yong Jiang, Pengjun Xie, Fei Huang, Shuai Wang, and Minhao Cheng. Lara: Benchmarking retrieval-augmented generation and long-context llms–no silver bullet for lc or rag routing. *arXiv preprint arXiv:2502.09977*, 2025a.

Kuan Li, Zhongwang Zhang, Huifeng Yin, Rui Ye, Yida Zhao, Liwen Zhang, Litu Ou, Dingchu Zhang, Xixi Wu, Jialong Wu, et al. Websailor-v2: Bridging the chasm to proprietary agents via synthetic data and scalable reinforcement learning. *arXiv preprint arXiv:2509.13305*, 2025b.

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

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

Zijian Li, Xin Guan, Bo Zhang, Shen Huang, Houquan Zhou, Shaopeng Lai, Ming Yan, Yong Jiang, Pengjun Xie, Fei Huang, et al. Webweaver: Structuring web-scale evidence with dynamic outlines for open-ended deep research. *arXiv preprint arXiv:2509.13312*, 2025e.

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

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

OpenAI. Introducing openai o3 and o4-mini, 2025b. URL https://openai.com/index/introducing-o3-and-o4-mini/.

OpenAI. Introducing simpleqa, 2025c. URL https://openai.com/index/introducing-simpleqa/.

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

Zile Qiao, Guoxin Chen, Xuanzhong Chen, Donglei Yu, Wenbiao Yin, Xinyu Wang, Zhen Zhang, Baixuan Li, Huifeng Yin, Kuan Li, et al. Webresearcher: Unleashing unbounded reasoning capability in long-horizon agents. *arXiv preprint arXiv:2509.13309*, 2025.

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

David Silver and Richard S Sutton. Welcome to the era of experience. *Google AI*, 1, 2025.

Liangcai Su, Zhen Zhang, Guangyu Li, Zhuo Chen, Chenxi Wang, Maojia Song, Xinyu Wang, Kuan Li, Jialong Wu, Xuanzhong Chen, Zile Qiao, Zhongwang Zhang, Huifeng Yin, Shihao Cai, Runnan Fang, Zhengwei Tao, Wenbiao Yin, et al. Scaling agents via continual pre-training, 2025.

Richard Sutton. The bitter lesson. *Incomplete Ideas (blog)*, 13(1):38, 2019.

Sijun Tan, Michael Luo, Colin Cai, Tarun Venkat, Kyle Montgomery, Aaron Hao, Tianhao Wu, Arnav Balyan, Manan Roongta, Chenguang Wang, Li Erran Li, Raluca Ada Popa, and Ion Stoica. rllm: A framework for post-training language agents. https://pretty-radio-b75.notion.site/rLLM-A-Framework-for-Post-Training-Language-Agents-21b81902c146819db63cd98a54ba5f31, 2025. Notion Blog.

Zhengwei Tao, Jialong Wu, Wenbiao Yin, Junkai Zhang, Baixuan Li, Haiyang Shen, Kuan Li, Liwen Zhang, Xinyu Wang, Yong Jiang, et al. Webshaper: Agentically data synthesizing via information-seeking formalization. *arXiv preprint arXiv:2507.15061*, 2025.

Kimi Team, Yifan Bai, Yiping Bao, Guanduo Chen, Jiahao Chen, Ningxin Chen, Ruijue Chen, Yanru Chen, Yuankun Chen, Yutian Chen, et al. Kimi k2: Open agentic intelligence. *arXiv preprint arXiv:2507.20534*, 2025.

Haoming Wang, Haoyang Zou, Huatong Song, Jiazhan Feng, Junjie Fang, Junting Lu, Longxiang Liu, Qinyu Luo, Shihao Liang, Shijue Huang, et al. Ui-tars-2 technical report: Advancing gui agent with multi-turn reinforcement learning. *arXiv preprint arXiv:2509.02544*, 2025.

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

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

Jialong Wu, Wenbiao Yin, Yong Jiang, Zhenglin Wang, Zekun Xi, Runnan Fang, Linhai Zhang, Yulan He, Deyu Zhou, Pengjun Xie, et al. Webwalker: Benchmarking llms in web traversal. *arXiv preprint arXiv:2501.07572*, 2025b.

Xixi Wu, Kuan Li, Yida Zhao, Liwen Zhang, Litu Ou, Huifeng Yin, Zhongwang Zhang, Yong Jiang, Pengjun Xie, Fei Huang, et al. Resum: Unlocking long-horizon search intelligence via context summarization. *arXiv preprint arXiv:2509.13313*, 2025c.

Xbench Team. Xbench-deepsearch, 2025. URL https://xbench.org/agi/aisearch.

An Yang, Anfeng Li, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chang Gao, Chengen Huang, Chenxu Lv, et al. Qwen3 technical report. *arXiv preprint arXiv:2505.09388*, 2025.

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

Qiying Yu, Zheng Zhang, Ruofei Zhu, Yufeng Yuan, Xiaochen Zuo, Yu Yue, Tiantian Fan, Gaohong Liu, Lingjun Liu, Xin Liu, et al. Dapo: An open-source llm reinforcement learning system at scale. *arXiv preprint arXiv:2503.14476*, 2025.

Aohan Zeng, Xin Lv, Qinkai Zheng, Zhenyu Hou, Bin Chen, Chengxing Xie, Cunxiang Wang, Da Yin, Hao Zeng, Jiajie Zhang, et al. Glm-4.5: Agentic, reasoning, and coding (arc) foundation models. *arXiv preprint arXiv:2508.06471*, 2025.

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

A Rollout Details

System Prompt

You are a deep research assistant. Your core function is to conduct thorough, multi-source investigations into any topic. You must handle both broad, open-domain inquiries and queries within specialized academic fields. For every request, synthesize information from credible, diverse sources to deliver a comprehensive, accurate, and objective response. When you have gathered sufficient information and are ready to provide the definitive response, you must enclose the entire final answer within **<answer></answer>** tags.

Tools

You may call one or more functions to assist with the user query.
You are provided with function signatures within **<tools></tools>** XML tags:
<tools>
{"type": "function", "function": {"name": "**search**", "description": "Perform Google web searches then returns a string of the top search results. Accepts multiple queries.", "parameters": {"type": "object", "properties": {"query": {"type": "array", "items": {"type": "string", "description": "The search query."}, "minItems": 1, "description": "The list of search queries."}}, "required": ["query"]}}}
{"type": "function", "function": {"name": "**visit**", "description": "Visit webpage(s) and return the summary of the content.", "parameters": {"type": "object", "properties": {"url": {"type": "array", "items": {"type": "string"}, "description": "The URL(s) of the webpage(s) to visit. Can be a single URL or an array of URLs."}, "goal": {"type": "string", "description": "The specific information goal for visiting webpage(s)."}}, "required": ["url", "goal"]}}}
{"type": "function", "function": {"name": "**PythonInterpreter**", "description": "Executes Python code in a sandboxed environment. To use this tool, you must follow this format:
1. The 'arguments' JSON object must be empty: {}.
2. The Python code to be executed must be placed immediately after the JSON block, enclosed within **<code>** and **</code>** tags.

IMPORTANT: Any output you want to see MUST be printed to standard output using the print() function.

Example of a correct call: <tool_call> {"name": "PythonInterpreter", "arguments": {}}
<code> import numpy as np # Your code here print(f"The result is: np.mean([1,2,3])") </code>
</tool_call>", "parameters": {"type": "object", "properties": {}, "required": []}}}
{"type": "function", "function": {"name": "**google_scholar**", "description": "Leverage Google Scholar to retrieve relevant information from academic publications. Accepts multiple queries. This tool will also return results from google search", "parameters": {"type": "object", "properties": {"query": {"type": "array", "items": {"type": "string", "description": "The search query."}, "minItems": 1, "description": "The list of search queries for Google Scholar."}}, "required": ["query"]}}} {"type": "function", "function": {"name": "**parse_file**", "description": "This is a tool that can be used to parse multiple user uploaded local files such as PDF, DOCX, PPTX, TXT, CSV, XLSX, DOC, ZIP, MP4, MP3.", "parameters": {"type": "object", "properties": {"files": {"type": "array", "items": {"type": "string"}, "description": "The file name of the user uploaded local files to be parsed."}}, "required": ["files"]}}}
</tools>
For each function call, return a json object with function name and arguments within **<tool_call></tool_call>** XML tags: **<tool_call>** {"name": <function-name>, "arguments": <args-json-object>} **</tool_call>**
Current date:

The above constitutes the system prompt of our ReAct rollout.

B Evaluation Details

For GAIA and WebWalkerQA, following the evaluation protocol of Li et al. (2025d), we adopt `Qwen2.5-72B-Instruct` as the judging model. The evaluation prompt is kept identical to that used in their work to ensure consistency and comparability. For xbench-DeepSearch and xbench-DeepSearch-2510, we adopt `Gemini-2.0-Flash-001` as the judge model. For BrowseComp and BrowseComp-ZH, we employ `GPT-4o-2024-08-06` as the judge model. For Humanity's Last Exam, we evaluate the 2,154 text-only questions following Chai et al. (2025). The evaluation prompt follows the official protocol, with the `o3-mini` serving as the evaluator. The evaluation prompt for these benchmarks is kept consistent with that described in the original paper to ensure alignment and reproducibility. The evaluation prompts used for each benchmark is provided in detail on our GitHub repository[5].

For general benchmarks, we adopt different evaluation strategies based on task type. For mathematical problems, since our system outputs a detailed report and datasets such as AIME25 and HMMT25 are relatively small in scale, we employ manual evaluation to ensure accuracy and fairness. For knowledge-based problems, we utilize the official evaluation script of SimpleQA to maintain consistency with established benchmarks.

C Post-training Synthetic Data Case

Question:
A military officer, who also served as governor in a western North American territory, commanded a mounted infantry unit during a period of significant mineral discovery in the region. His official report on the discovery prompted the minting of a special commemorative coin in a certain year in the mid-19th century. During that same year, the unit he commanded was involved in a military conflict against a neighboring country. Just over a decade later, this unit was officially redesignated and would be assigned to a new division in the early 1920s. In the 1930s, this redesignated regiment was involved in an organizational swap. Which other regiment was it exchanged for?
Answer:
12th Cavalry Regiment

Question:
An 18th-century travelogue, later adapted for a radio series, describes a port town in southeastern England as notable for its rampant illicit trade. This town was also the home of a 16th-century gentleman whose murder led to his wife's execution. Centuries later, another resident of the same town was granted letters patent providing special commercial privileges in a particular year of the early 19th century. During that same year, a collector, whose large collection of manuscript poems was later auctioned, secured a patent for a method of grinding inks. In that year, a patent of nobility was issued to a German family; what is the German term for the princely status it conferred?
Answer:
Fürstenstand

Question:
In trisilylamine ($N(SiH_3)_3$), the Si-N bond length is 1.736 Å. Substituting one silyl group with methyl to form $(CH_3)N(SiH_3)_2$ elongates the Si-N bond to 1.752 Å. Calculate the percentage increase in bond length due to diminished hyperconjugation, and identify which specific orbital interaction weakens most significantly. Use covalent radii: Si=1.11 Å, N=0.70 Å, C=0.77 Å.
Answer:
$n \rightarrow \sigma^*_{Si-C}$

The first two cases above are synthetically generated high-quality, high-uncertainty, superhuman question–answer pairs, examples of a caliber that is exceptionally difficult to produce via human annotation. The third case represents a PhD-level research question, demanding deep domain expertise, multi-step reasoning.

[5] https://github.com/Alibaba-NLP/DeepResearch/tree/main/evaluation

D Environment Details

We utilize five tools for Tongyi DeepResearch, namely Search, Visit, Python Interpreter, Google Scholar, and File Parser[6]:

- **Search** leverages the Google search engine for information retrieval. The tool accepts a list of one or more search queries to be executed concurrently. For each query, it returns the top-10 ranked results, with each result comprising a title, a descriptive snippet, and its corresponding URL.
- **Visit** is designed for targeted information extraction from web pages. The tool takes as input a set of web pages, where each page is paired with a dedicated information-seeking goal. The process begins by employing Jina (Jina.ai, 2025) to parse the full content of a given web page. Subsequently, a summary model processes this content to extract only the information pertinent to that page's specific goal.
- **Python Interpreter** is used to execute Python code within a sandboxed environment. The input is a string of Python code, which must be enclosed within <*code*> tags for proper execution. The tool runs the provided code and captures its standard output; therefore, any results or values intended to be seen must be explicitly passed to the *print()* function. This capability enables dynamic computation, data manipulation, and the use of various Python libraries in a secure and isolated manner.
- **Google Scholar** is used to retrieve information from academic publications. The input consists of a list of one or more search queries, allowing for multiple, distinct searches within a single tool call. The tool leverages the Google Scholar search engine to execute each query and gather relevant scholarly literature, such as articles, papers, and citations.
- **File Parser** answers user queries by analyzing a mix of documents, web pages, and multimedia files (*e.g.*, PDF, DOCX, MP4) from local or URL sources. It works in two steps: first, it converts all input into plain text, transcribing audio/video content when necessary. Second, a summary model reads this unified text to generate a direct answer to the user's question

[6]Since our system relies on several internal APIs and fallback strategies (as described in Section 3.4.3), we provide alternative open implementations in our open-source GitHub repository to facilitate public use. We have verified through extensive testing that these substitutions can faithfully reproduce our results.