PARALLELMUSE: Agentic Parallel Thinking for Deep Information Seeking

Baixuan Li[†][(✉)], Dingchu Zhang[†], Jialong Wu[†], Wenbiao Yin[(✉)], Zhengwei Tao, Yida Zhao,
Liwen Zhang, Haiyang Shen, Runnan Fang, Pengjun Xie, Jingren Zhou, Yong Jiang[(✉)]

Tongyi Lab ✦ , Alibaba Group

🔗 https://tongyi-agent.github.io/blog
🐙 https://github.com/Alibaba-NLP/DeepResearch

Abstract

Parallel thinking expands exploration breadth, complementing the deep exploration of information-seeking (IS) agents to further enhance problem-solving capability. However, conventional parallel thinking faces two key challenges in this setting: inefficiency from repeatedly rolling out from scratch, and difficulty in integrating long-horizon reasoning trajectories during answer generation, as limited context capacity prevents full consideration of the reasoning process. To address these issues, we propose **PARALLELMUSE**, a two-stage paradigm designed for deep IS agents. The first stage, *Functionality-Specified Partial Rollout*, partitions generated sequences into functional regions and performs uncertainty-guided path reuse and branching to enhance exploration efficiency. The second stage, *Compressed Reasoning Aggregation*, exploits reasoning redundancy to losslessly compress information relevant to answer derivation and synthesize a coherent final answer. Experiments across multiple open-source agents and benchmarks demonstrate up to **62%** performance improvement with a **10–30%** reduction in exploratory token consumption.

1 Introduction

Deep information-seeking (IS) agents[1] (OpenAI, 2025b; Team, 2025a;b) can actively uncover hard-to-access information, extending large language models beyond static training data and empowering them to reason over real-world knowledge. This capability emerges from a continual loop of environmental[2] interaction and internal reasoning, through which the agent incrementally builds reasoning depth within a single execution to effectively solve complex problems (Wu et al., 2025c;a; Li et al., 2025c; Tao et al., 2025; Li et al., 2025b). In this setting, parallel thinking provides a natural form of test-time scaling: by expanding the number of parallel exploration paths, it broadens the agent's search while maintaining reasoning depth along each path, thereby enhancing overall performance without altering model parameters.

As commonly recognized, parallel thinking can be viewed as a two-stage process (Li et al., 2025a), involving an initial stage of exploratory sampling and a subsequent stage dedicated to answer generation across sampled candidates. In this work, we extend this paradigm to the setting of deep IS agents,

---

[†]Equal contribution.

[✉]Correspondence to: baixuan@seu.edu.cn, {yinwenbiao.ywb, yongjiang.jy}@alibaba-inc.com.

[1]The agents discussed in this work are function-calling agents that adhere to the standard ReAct (Yao et al., 2023) paradigm, operating through an iterative think → tool call loop.

[2]This work focuses on deep information-seeking agents, where the term "environment" specifically refers to the web environment or information sources with which the agent interacts.

referred to as **PARALLELMUSE**. We analyze how the characteristics of each stage manifest under agentic conditions and propose systematic optimization strategies derived from these pilot observations.

First, in the exploratory sampling stage, conventional rollout strategies in parallel thinking typically restart from scratch at each iteration, resampling the entire exploration space (AI, 2025; Fu et al., 2025; Zeng et al., 2025). During certain reasoning phases, however, exploration diversity is inherently low, making repeated rollouts inefficient and token-expensive. Prior work introduces *partial rollout* methods that estimate exploration potential via uncertainty and selectively branch where uncertainty is high (Hou et al., 2025; Dong et al., 2025; Li et al., 2025e), but these approaches assume functional homogeneity across tokens, implying that all tokens contribute equally to exploration and exhibit similar uncertainty.

This assumption holds in purely reasoning-oriented tasks such as mathematics or coding but fails in agentic IS settings, where the model must generate both reasoning and tool-call actions. These behaviors naturally form distinct *functional regions* with different uncertainty patterns. Motivated by this observation, we propose the ***Functionality-Specified Partial Rollout*** method as the first stage of the PARALLELMUSE framework. The method segments the generated sequence into functional regions, estimates uncertainty independently within each, and selectively expands rollouts for reasoning steps with higher exploration potential. This enables behavior-level estimation of exploration potential, allowing targeted exploration across different functional behaviors, and improving overall efficiency in agentic tasks.

Second, in the answer generation stage, parallel thinking produces multiple reasoning candidates from which a single answer must be derived, typically through *answer selection* (Wang et al., 2022; Fu et al., 2025) or *answer aggregation* (Jiang et al., 2023; Liang et al., 2024; Zhang et al., 2025b; Qiao et al., 2025). In complex agentic tasks with vast sampling spaces, the correct answer may not dominate numerically, as it often constitutes only a small fraction of all possible sampled outcomes. Moreover, the continual incorporation of external, non–model-generated information shifts the output distribution, further hindering confidence calibration (Jang et al., 2024). As a result, majority voting (Wang et al., 2022) and confidence-based selection (Fu et al., 2025) often fail. For answer aggregation, focusing only on final answers neglects intermediate reasoning, while incorporating entire reasoning traces is infeasible for long-horizon agents due to context limits. Recent work (Qiao et al., 2025) seeks a compromise by aggregating only the last few reasoning steps, but this discards earlier content, which often reflects planning and problem decomposition and is essential for evaluating the coherence of the final answer.

To address this challenge, we first conceptualize the IS task as a process of discovering key entities and building connections among them (Li et al., 2025c; Tao et al., 2025; Li et al., 2025b). Based on our preliminary observations, only a small portion of the entities explored by an agent contribute meaningfully to the final answer, revealing substantial redundancy and strong potential for lossless compression in the generated interaction–reasoning trajectories. Building on this insight, we propose the ***Compressed Reasoning Aggregation*** method as the second stage of the PARALLELMUSE paradigm. The method first condenses all reasoning candidates into concise, structured reports that preserve only information relevant to answer derivation, and then aggregates these compressed reports to produce the final answer. This approach enhances processing efficiency and mitigates the bias of majority-based selection, enabling more reliable and coherent answer generation.

The proposed PARALLELMUSE is evaluated on four open-source deep IS agents, including GPT-OSS-20B (OpenAI, 2025a), GPT-OSS-120B, DeepSeek-V3.1-Terminus (Liu et al., 2024), and Tongyi-DeepResearch-30B-A3B (Team, 2025b), across four challenging benchmarks that jointly assess deep search and reasoning abilities: BrowseComp (Wei et al., 2025), BrowseComp-zh (Zhou et al., 2025), GAIA (Mialon et al., 2023), and Humanity's Last Exam (HLE) (Phan et al., 2025). Extensive experiments show that PARALLELMUSE achieves up to **62%** improvement while requiring only **70–90%** of the exploratory token cost of conventional parallel thinking. Beyond the empirical gains, our analysis provides key insights into the mechanisms of deep IS agents, offering guidance for future research in agentic reasoning.

2 Pilot Observation

We begin with a preliminary analysis of the characteristics of deep information-seeking (IS) agents and their associated tasks, providing insights from two perspectives: (i) exploratory sampling (trajectory rollout) and (ii) the resulting interaction–reasoning trajectories.

2.1 Distinct Uncertainty Patterns Across Functional Reasoning Steps

In deep IS tasks, agents must not only reason over internal knowledge but also explore unknown information through tool use and environmental interaction. While pure reasoning models use tokens exclusively for internal reasoning, deep IS agents additionally allocate tokens for tool invocation to retrieve external information, reflecting distinct functional roles in token utilization.

Formally, each step consists of a reasoning segment, a tool invocation, and its corresponding tool response. We denote the set of tokens generated by the model at step t as $\mathcal{T}_t = \{x_{t,1}, x_{t,2}, ..., x_{t,m}\}$, with $x_{t,i}$ denoting the i-th token. The set is partitioned into two subsets: $\mathcal{T}_t^r$, representing *reasoning* tokens, and $\mathcal{T}_t^e$, representing *exploration* tokens. This partition holds for each step, implying $\mathcal{T}_t^r \cup \mathcal{T}_t^e = \mathcal{T}_t$. By extension, aggregating these sets across the entire trajectory yields global sets $(\mathcal{T}, \mathcal{T}^r, \mathcal{T}^e)$. In contrast, a pure reasoning task would have an empty exploration set, $\mathcal{T}^e = \varnothing$, satisfying $\mathcal{T} = \mathcal{T}^r$.

Furthermore, we observe that the uncertainty associated with tokens in the $\mathcal{T}^r$ and $\mathcal{T}^e$ subsets exhibits distinct temporal dynamics during the agentic interaction-reasoning process. To quantitatively capture this behavior, we use the **perplexity** (PPL) of each reasoning step, which is defined as the average PPL of tokens within step t, as a proxy for the deep IS agent's self-uncertainty.

$$\text{PPL}(f, t) = \exp\left(-\frac{1}{|\mathcal{T}_t^f|}\sum_{i=0}^{|\mathcal{T}_t|}\log p\left(x_{t,i} \mid x_{<t,i}\right)\right), \quad x_{t,i} \in |\mathcal{T}_t^f|, \quad f \in \{r, e\}, \tag{1}$$

where f represents the functional region of the entire trajectory, which is partitioned into a reasoning region r and an exploration region e.
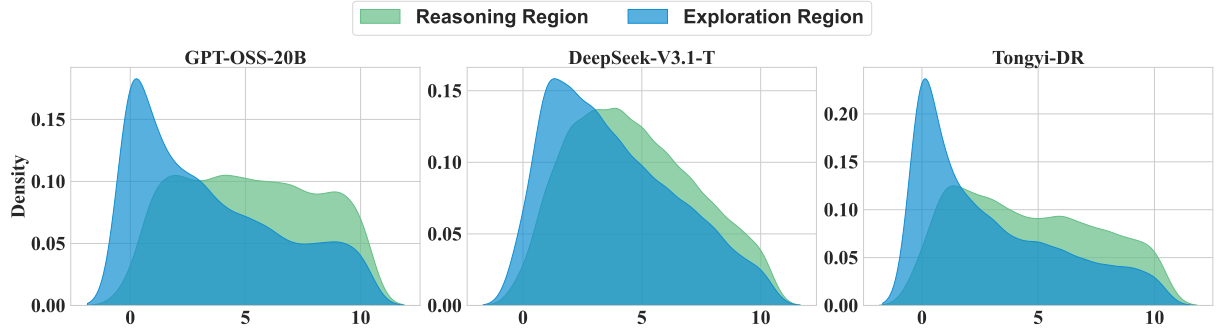

Figure 1: KDE-smoothed distribution of steps with top-4 uncertainty on the BrowseComp subset (truncated to earlier steps as later ones are typically more certain). DeepSeek-V3.1-T denotes DeepSeek-V3.1-Terminus, and Tongyi-DR denotes Tongyi-DeepResearch-30B-A3B.

We analyze $\text{PPL}(r, t)$ and $\text{PPL}(e, t)$ across steps to characterize the distinct uncertainty dynamics of reasoning and exploration within the agentic reasoning-interaction process. As shown in Figure 1, across multiple deep IS models, we examine the distribution of the top-4 uncertainty steps observed during task execution. The results reveal a consistent pattern: exploration uncertainty reaches its highest levels at the earliest stages, when no external information has yet been gathered, while reasoning uncertainty peaks slightly later as the agent begins integrating retrieved information into its internal reasoning process.

Specifically, *exploration uncertainty* tends to peak at the beginning of the task, when the agent has not yet gathered external information and must explore the environment under minimal prior knowledge.

Reasoning uncertainty, in contrast, reaches its peak slightly later—still in the early stage—when the agent begins to process and integrate the newly retrieved information into its internal reasoning chain. As the task proceeds, both forms of uncertainty gradually decline as the agent accumulates knowledge and its reasoning process becomes more grounded, resulting in increasingly confident decisions and actions.

This observation further informs the design of the **Functionality-Specified Partial Rollout** method in PARALLELMUSE, which enhances agentic parallel thinking by enabling more efficient exploration.

2.2 From Exploration Redundancy to Losslessly Compressible Trajectory

Following recent studies (Li et al., 2025c; Tao et al., 2025; Li et al., 2025b), deep IS tasks can be formulated as a process of *entity discovery* and *relation construction*. Formally, given an initial query or objective q, the agent incrementally builds a set of discovered entities $\mathcal{V} = \{v_1, v_2, \ldots, v_N\}$ through iterative interactions with external information sources. At each step t, the agent performs exploration to retrieve candidate entities $\tilde{\mathcal{V}}_t$ and reasoning to determine their relevance and relational connections. The evolving information state of the agent can thus be expressed as:

$$\mathcal{G}_t = (\mathcal{V}_t, \mathcal{R}_t), \tag{2}$$

where $\mathcal{V}_t$ denotes the set of *effective entities* (i.e., entities considered valid for answer derivation) and $\mathcal{R}_t \subseteq \mathcal{V}_t \times \mathcal{V}_t$ represents the relations constructed among them. The goal of the task is to iteratively refine $\mathcal{G}_t$ until it contains the key entities and connections necessary for deriving the final answer. Once the entire agentic reasoning process terminates, the final graph $\mathcal{G}_{\text{final}} \supseteq \mathcal{I}_{\text{answer}}$ (where $\mathcal{I}_{\text{answer}}$ denotes all information essential for answer derivation), serving as the core representation of the reasoning trajectory.

Based on this formulation of deep IS tasks, we approximate the redundancy of a task's complete reasoning trajectory by measuring the proportion of *effective entities*—those that contribute directly or indirectly to answer derivation—within all entities discovered during execution. Formally, let $\mathcal{V}_{\text{total}}$ denote the set of all entities explored by the agent during a task, and let $\mathcal{V}_{\text{eff}} \subseteq \mathcal{V}_{\text{total}}$ represent the subset of entities that are directly or indirectly useful for deriving the final answer. The redundancy ratio Γ_{red} is defined as:

$$\Gamma_{\text{red}} = 1 - \frac{|\mathcal{V}_{\text{eff}}|}{|\mathcal{V}_{\text{total}}|}. \tag{3}$$

A higher Γ_{red} indicates greater redundancy in the agent's interaction–reasoning trajectory, which can be interpreted as a stronger potential for *lossless compression* of the reasoning process. In this context, lossless compression refers to reducing redundant entities and reasoning steps while preserving all reasoning information necessary for the complete derivation of the final answer (i.e., extracting and representing $\mathcal{G}_{\text{final}}$), and thus Γ_{red} can be regarded as an approximate indicator of the degree of lossless compressibility.

Accordingly, we compute the reasoning trajectory redundancy of several mainstream deep IS agents during real task execution. As illustrated in Figure 2, all models exhibit consistently high redundancy, indicating that the reasoning trajectories in deep IS tasks are highly losslessly compressible. This observation supports the design of the **Compressed Reasoning Aggregation** method in PARALLELMUSE, which aims to integrate as much effective reasoning information as possible into final aggregation with minimal information loss.
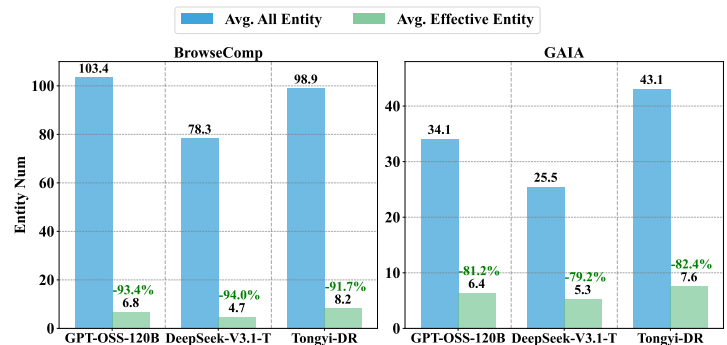

Figure 2: Average entity count per task and per model, where entities are extracted by GPT-4.1 based on the complete reasoning trajectory and ground-truth answer.
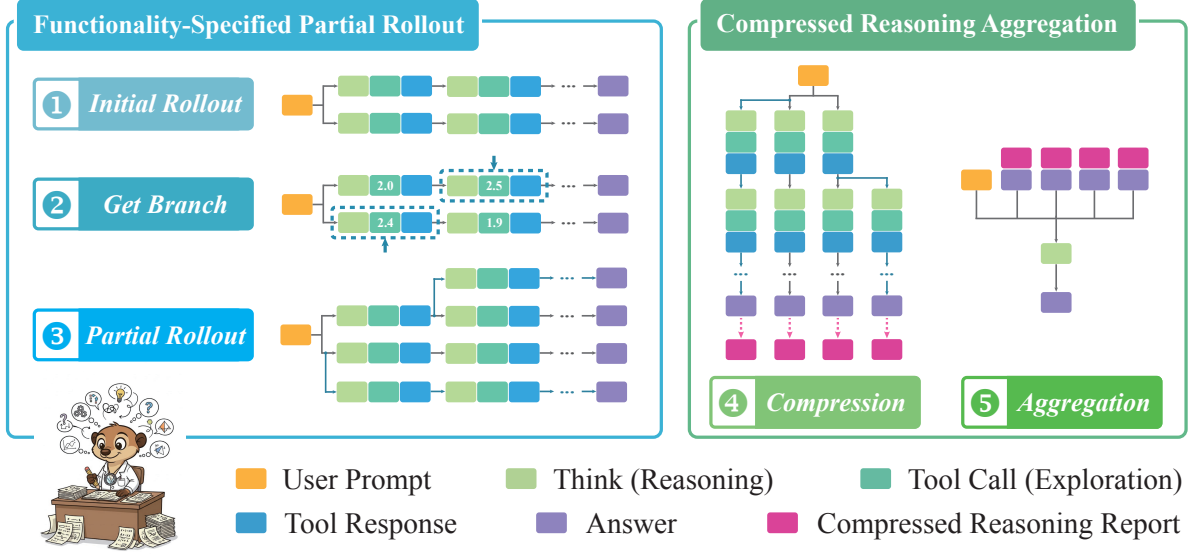
Figure 3: Workflow of PARALLELMUSE, including (*Left*) the Functionality-Specified Partial Rollout, where the *Get Branch* shows the selection of top-*k* steps based on (exploration) tool-call uncertainty (just as an example of branching criterion), and (*Right*) the Compressed Reasoning Aggregation.

3 ParallelMuse

The proposed PARALLELMUSE is a two-stage agentic parallel thinking paradigm comprising two complementary components: (i) *Functionality-Specified Partial Rollout* and (ii) *Compressed Reasoning Aggregation*. As shown in Figure 3, these correspond to the first stage of exploratory sampling and the second stage of answer generation in the overall parallel thinking process, respectively.

3.1 Functionality-Specified Partial Rollout

Functionality-Specified Branching Step Identification. Agent models inherently partition generated tokens into functional regions, typically reasoning and exploration, signaled by special tokens such as <think> and <tool_call>. We leverage these markers to identify distinct functional segments within the generation process. To enable more targeted partial rollout, it is essential to identify reasoning steps with higher exploration potential. We measure this potential through the model's generation uncertainty at each step, as higher uncertainty indicates greater diversity in possible continuations and thus a broader exploration space. Accordingly, we compute the **step-level perplexity** (PPL) within each functional region, as defined in equation 1, to quantify the agent's generation uncertainty.

This process is conducted in an offline manner to ensure optimal branch selection. As shown in Figure 3 (*Left*), we first generate M initial trajectories from scratch, compute step-level PPL for reasoning and exploration regions along these trajectories for each step, and select the top-*k* steps with the highest uncertainty in the chosen functional region $f \in \{r, e\}$ (defined in equation 1) as branching points for subsequent partial rollouts. It is noted that M, k, and f are tunable hyper-parameters.

Asynchronous Partial Rollout. From the selected high-uncertainty steps, $N - M$ additional partial rollouts are launched asynchronously to expand exploration, where N is the overall sampling budget. Each branch directly reuses the preceding context rather than regenerating it from scratch, continuing from cached hidden states in the key–value (KV) cache (Li et al., 2024; Wu et al., 2025b). This reuse eliminates redundant forward passes, yielding substantial savings in both token and compute cost.

We implement an asynchronous rollout engine to parallelize branch generation while preserving each branch's causal decoding consistency, enabling multiple branches to expand concurrently and efficiently.

The acceleration comes from two sources: (i) *prefix reuse* via KV caching and (ii) *asynchronous parallelization*. Let branch j reuse a prefix of token length p_j and generate a suffix of token length s_j. With cold decoding (no KV reuse), the cost is $C_j^{\text{cold}} = c_{\text{cold}} \cdot (p_j + s_j)$; with KV reuse, the cost is $C_j^{\text{hot}} = c \cdot s_j$. Here, $c > 0$ denotes the per-token compute cost under cached decoding (with KV reuse), and $c_{\text{cold}} \geq c$ denotes the per-token cost when regenerating from scratch (without KV reuse).

$$\text{ReuseFactor} \equiv \frac{\sum_j C_j^{\text{cold}}}{\sum_j C_j^{\text{hot}}} = \frac{c_{\text{cold}}}{c} \left(1 + \frac{\sum_j p_j}{\sum_j s_j} \right). \tag{4}$$

Asynchronous scheduling parallelizes hot decoding across P active branches. If $\alpha \in [0, 1]$ denotes the parallelizable ratio, the throughput gain obeys the Amdahl-type bound (Amdahl, 1967):

$$\text{ParaFactor}(P) \leq \frac{1}{(1 - \alpha) + \alpha/P}. \tag{5}$$

Combining equation 4 and equation 5 yields the overall speedup: $\text{Speedup}_{\text{total}} \lesssim \text{ReuseFactor} \times \text{ParaFactor}(P)$. In the practical regime with efficient KV caching ($c \approx c_{\text{cold}}$), high parallelizability ($\alpha \approx 1$), and P within hardware concurrency, this simplifies to

$$\text{Speedup}_{\text{total}} \approx \left(1 + \frac{\sum_j p_j}{\sum_j s_j} \right) P. \tag{6}$$

This design jointly exploits deterministic prefix reuse and asynchronous parallelization to achieve near-linear speedup in exploration efficiency with relatively lower token cost.

3.2 Compressed Reasoning Aggregation

Structured Report-Style Compression. Building on the observations in Section 2.2, we note that the complete reasoning trajectories obtained after the first-stage exploratory sampling in deep information-seeking (IS) tasks exhibit high redundancy, implying strong lossless compressibility with respect to answer derivation. To effectively integrate richer intermediate reasoning information during the answer aggregation stage while maintaining computational efficiency and avoiding context overflow, we first compress each candidate reasoning trajectory produced in the exploratory stage.

As shown in Figure 3 (*Right*), for each reasoning trajectory generated in solving deep IS tasks, the compression objective is to produce a structured report that preserves key elements essential to answer derivation. The report records:

(i) **Solution Planning:** describes how the main problem is decomposed into subproblems, including their dependency structure and execution order.

(ii) **Solution Methods:** specifies the tools invoked to solve each subproblem, the corresponding parameters used, and any subanswers that contribute directly or indirectly to the final solution.

(iii) **Final Reasoning:** illustrates how the identified subproblems and associated subanswers are integrated to derive the final answer.

Irrelevant exploratory content, including redundant tool responses and ineffective reasoning or tool calls, is removed. This process effectively reconstructs the agent's internal information state graph $\mathcal{G}$ (defined in equation 2), which comprehensively captures all information relevant to answer derivation.

Reasoning-Guided Answer Aggregation. After obtaining N compressed reports from the exploratory sampling stage, we can jointly consider all N globally compressed reasoning candidates within the limited context window, rather than focusing only on their final answers or partial reasoning traces. This enables a more comprehensive evaluation of reasoning coherence and supports a more reliable determination of the optimal answer. In this aggregation stage, we explicitly prevent the model from

relying solely on answer consistency as a correctness signal, mitigating the bias toward majority answers and ensuring that reasoning coherence remains the primary criterion. Furthermore, we restrict the model from trivially concatenating or enumerating different answers to preserve aggregation validity.

It is also important to note that each report already contains sufficient tool-calling provenance and attribution information for answer derivation. Therefore, during the aggregation phase, the model does not perform additional tool invocations for secondary verification but instead conducts reasoning purely over the information encoded in the N reports. Empirical results later demonstrate that this approach is both effective and computationally efficient.

4 Experiments

We focus on evaluating the effectiveness and efficiency of applying PARALLELMUSE to existing deep information-seeking (IS) agents. Comprehensive experiments are conducted to examine the impact of its two stages both individually and jointly, assessing how each contributes to overall task performance.

4.1 Setup

Benchmarks. We evaluate PARALLELMUSE on four challenging deep IS benchmarks: BrowseComp (Wei et al., 2025), BrowseComp-zh (Zhou et al., 2025), GAIA (Mialon et al., 2023), and Humanity's Last Exam (HLE) (Phan et al., 2025). These benchmarks jointly assess both deep search and reasoning capabilities, with BrowseComp and BrowseComp-zh placing greater emphasis on deep search, HLE focusing more on reasoning, and GAIA providing a balanced evaluation across both dimensions.

For efficient text-only evaluation, we use sampled subsets from large-scale datasets: 200 randomly selected tasks from BrowseComp, 157 search-focused text-only tasks from HLE, and 103 text-only tasks from GAIA (Li et al., 2025d), while using the full 289-task set for BrowseComp-zh.

Tools. We adopt the standard tool configuration commonly used in deep IS agents (Wu et al., 2025a; Li et al., 2025c; Tao et al., 2025; Li et al., 2025b; Qiao et al., 2025), which includes two core tools for interacting with the web environment and retrieving external information:

- **Search:** Performs batched Google queries and returns the top-10 ranked results for each.
- **Visit:** Fetches webpages from multiple URLs and extracts information relevant to the given goal.

Agent Models. We select four open-source agent models with diverse parameter scales and advanced tool-use capabilities for deep IS tasks: GPT-OSS-20B (OpenAI, 2025a), GPT-OSS-120B, DeepSeek-V3.1-Terminus (DeepSeek-V3.1-T, 671B) (Liu et al., 2024), and Tongyi-DeepResearch-30B-A3B (Tongyi-DR-30B-A3B) (Team, 2025b). All agent models are invoked under the official function-calling protocol. Unless otherwise specified, we use the *same* agent model to perform both stages of the PARALLELMUSE.

Baselines. In addition to the standard inference baseline without any parallel thinking (*No Scaling*), we compare PARALLELMUSE against several mainstream parallel thinking baselines. These include: (i) Self-Consistency (*Majority Vote*) (Wang et al., 2022), which selects the most frequent answer across multiple trajectories as the final output; (ii) *Max #Tool Call* (Zeng et al., 2025), which heuristically chooses the answer derived from the trajectory with the largest number of environment interactions; and (iii) DeepConf (*Weighted Vote*) (Fu et al., 2025), which weights answers by the model's confidence over each trajectory and selects the answer with the highest final score.

Table 1: Default settings of our proposed PARALLELMUSE.

Hyper-Parameters	Values
Sampling Budget N	8
#Initial Rollout M	1
Branching PPL Top-K	2
#Branching Times per Step	3

Evaluation Metrics and Hyper-Parameters. All evaluations are performed under the LLM-as-a-Judge paradigm (Gu et al., 2024),

using the official evaluation prompts and judging models specified by each benchmark's released configuration. For the *No Scaling* method, we report the average pass rate over N independent rollouts, while for parallel thinking methods, which yield a single final answer from N rollouts, we report the pass rate of that final output. For our proposed PARALLELMUSE, unless otherwise specified, the default hyper-parameter settings are listed in Table 1. To ensure fair comparison and reproducibility, all agent model–specific hyper-parameters are aligned with their official optimal configurations for tool usage.

4.2 Overall Performance

Table 2: Overall performance. Scores marked with ‡ represent full-benchmark results, whereas unmarked scores correspond to our benchmark settings. Both PARALLELMUSE and other parallel thinking baselines are evaluated under the default configurations as described in Section 4.1. The specific strategy for selecting functional regions in PARALLELMUSE's partial rollout is discussed in Section 4.3.

Model / Framework	Method	BrowseComp	BrowseComp-zh	GAIA	HLE
Closed-Source Deep Information-Seeking Agents					
Claude-4-Sonnet	No Scaling	12.2‡	29.1	68.3	20.3‡
OpenAI-o3	No Scaling	49.7‡	58.1	70.5	26.6‡
Kimi Researcher	No Scaling	–	–	–	26.9‡
OpenAI DeepResearch	No Scaling	51.5‡	42.9	67.4	26.6‡
ChatGPT Agent	No Scaling	68.9‡	–	–	41.6‡
Open-Source Deep Information-Seeking Agents					
GPT-OSS-20B	No Scaling	30.9	28.6	63.4	24.2
	Majority Vote	44.0	38.8	69.9	24.2
	Max #Tool Call	17.0	19.0	58.3	26.1
	Weighted Vote	41.0	37.0	68.9	31.2
	PARALLELMUSE	**49.0**	**44.3**	**72.8**	**32.5**
GPT-OSS-120B	No Scaling	34.9	33.8‡	36.0	74.3	36.3
	Majority Vote	48.5	46.7	77.7	43.3
	Max #Tool Call	17.5	26.3	68.9	36.9
	Weighted Vote	48.0	45.7	82.5	45.2
	PARALLELMUSE	**56.5**	**54.3**	**85.4**	**45.9**
DeepSeek-V3.1-T	No Scaling	23.2	36.1	61.0	25.0 | 21.7‡
	Majority Vote	30.0	45.0	70.9	26.1
	Max #Tool Call	17.5	28.0	57.3	27.4
	Weighted Vote	29.5	45.0	70.9	28.0
	PARALLELMUSE	**39.0**	**50.2**	**74.8**	**37.6**
Tongyi-DR-30B-A3B	No Scaling	51.0 | 43.4‡	45.3	73.6	38.5 | 32.9‡
	Majority Vote	60.0	56.8	77.7	40.1
	Max #Tool Call	41.0	36.3	75.7	38.2
	Weighted Vote	62.0	53.6	78.6	42.7
	PARALLELMUSE	**65.0**	**57.1**	**79.6**	**52.2**

We report the performance of closed-source deep IS agents across all benchmarks and compare them with open-source agents equipped with our proposed PARALLELMUSE and several representative parallel thinking baselines. As shown in Table 2, PARALLELMUSE consistently achieves the highest performance gains over all baselines across every agent model and benchmark. Notably, when applied to Tongyi-DR-30B-A3B, it attains performance comparable to or surpassing that of most closed-source agents.

It is important to note that we further observe that the *Weighted Vote*, which relies on self-estimated confidence, underperforms *Majority Vote* across all models except Tongyi-DR-30B-A3B. This can be attributed to confidence miscalibration in agentic settings: as agents repeatedly integrate external, non–model-generated content (e.g., tool responses), their internal probability distributions shift, degrading the reliability of confidence scores (Jang et al., 2024; Chhikara, 2025). The exception is the HLE benchmark, which emphasizes reasoning with limited external interaction, and Tongyi-DR-30B-A3B, which benefits from continual pre-training (Su et al., 2025) that improves calibration over Search and Visit tool responses.

In contrast, our proposed PARALLELMUSE avoids confidence-based answer selection altogether, mitigating this source of bias and yielding consistent and substantial improvements across all agent models.

4.3 Analysis of Partial Rollout over Distinct Functional Regions

Table 3: Performance comparison between full from-scratch rollouts and partial rollouts guided by functional-region uncertainty. Detailed configurations are listed in Table 1.

Agent Model	Functional Region	BrowseComp	BrowseComp-zh	GAIA	HLE
GPT-OSS-120B	From Scratch: No Region	34.9	36.0	74.2	36.3
	Partial: Reasoning	37.9	**43.1**	76.1	36.3
	Partial: Exploration	**39.9**	41.6	**77.9**	**37.5**
	Partial: Mixed	38.1	42.9	76.7	37.1
DeepSeek-V3.1-T	From Scratch: No Region	23.2	36.1	**61.0**	25.0
	Partial: Reasoning	**26.5**	**39.8**	60.2	**26.4**
	Partial: Exploration	23.8	37.9	60.6	25.3
	Partial: Mixed	23.4	38.1	60.3	25.1

To analyze the effect of identifying branching steps based on different functional regions in partial rollout, we report the average pass rate after 8 rollouts, as shown in Table 3. The *From Scratch* setting denotes full rollouts without reusing context, where functional region selection is not applicable. For our proposed PARALLELMUSE[3], we evaluate three functionality-specified partial rollout strategies: using uncertainty from the *Reasoning* region only, using uncertainty from the *Exploration* region only, and a *Mixed* configuration where half of the top-2 branching steps are selected from each region. The *Mixed* setting acts as a compromise, integrating uncertainty cues from both regions.

The results show that the effectiveness of branching based on different functional-region uncertainties varies across models, reflecting their inherent behavioral and capability differences. For instance, GPT-OSS-120B benefits less from reasoning-based branching, as its strong adaptive reasoning mechanism already yields consistently high-quality reasoning with limited exploration potential. In contrast, DeepSeek-V3.1-T employs function calling outside of the `thinking` mode, resulting in weaker internal reasoning capacity and thus higher sampling potential in reasoning steps. These insights heuristically inform our choice of functional-region uncertainty for partial rollout in PARALLELMUSE.

We also observe that partial rollout consistently outperforms full from-scratch rollout in most cases. This improvement arises from more targeted exploration. In deep IS tasks, where interaction with the web environment induces an extremely large exploration space, unguided rollouts struggle to identify effective search paths and often fall into local optima. In contrast, uncertainty-guided partial rollout functions analogously to Monte-Carlo Tree Search (MCTS) (Browne et al., 2012): while MCTS reuses high-reward trajectories, PARALLELMUSE reuses low-uncertainty (low-potential) paths and selectively

---

[3]We omit results of partial rollout without functional-region distinction (i.e., treating all tokens as homogeneous), as our preliminary experiments show that this setting performs comparably to full from-scratch rollouts and provides no observable gains from branching at high-uncertainty steps.

expands exploration at high-uncertainty steps. This strategy allocates the limited sampling budget toward regions with greater expected exploration gain, enhancing both efficiency and effectiveness.

4.4 Performance Gains from Compressed Reasoning Aggregation

In Section 4.3, we examined the performance gains arising from the first-stage partial rollout of the proposed PARALLELMUSE. In this section, we isolate and analyze the effectiveness of its second-stage answer aggregation method, independently assessing its contribution to overall performance.

As shown in Figure 4, even without the sampling (exploration) gains from the first-stage partial rollout, the proposed *Compressed Reasoning Aggregation* (the second-stage of PARALLELMUSE) alone yields the most significant improvement. Notably, this approach performs near-lossless compression over each agentic reasoning trajectory to efficiently integrate reasoning information without invoking additional tool calls
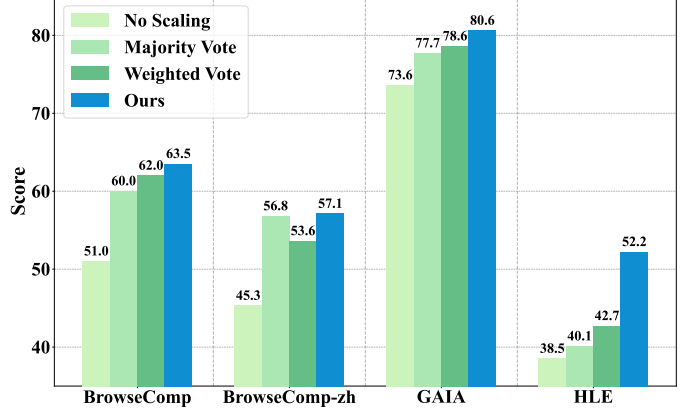
Figure 4: Performance gains from different answer generation methods, with sampling fixed to 8 from-scratch rollouts to isolate sampling (exploration) effects.

for secondary verification. By maximizing the exploitation of existing sampled information during aggregation, it achieves a balanced improvement in both efficiency and solution quality.

4.5 Efficiency Gains through Context Reuse and Trajectory Compression
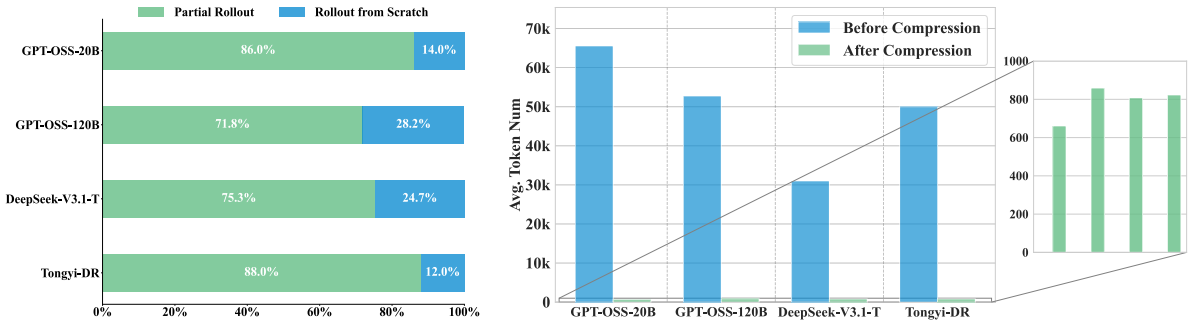
Figure 5: Efficiency gains using PARALLELMUSE. (i) (*Left*) Token reduction through context reuse in our partial rollout method. We take the token consumption per trajectory of the from-scratch rollout as the baseline. The green bars represent the token cost after applying partial rollout (the numbers above indicate the ratio relative to the baseline), while the remaining blue bars show the proportion of tokens saved. (ii) (*Right*) Comparison of context token usage before and after trajectory compression.

We conduct a detailed analysis of the efficiency gains achieved by our proposed PARALLELMUSE, which primarily arise from two complementary sources:

(i) **Token reduction via context reuse in *Functionality-Specified Partial Rollout*.** As shown in Figure 5 (*Left*), our method (Partial Rollout) achieves up to **28%** token savings by effectively reusing context instead of regenerating it from scratch (Rollout from Scratch). The efficiency gain increases with sampling scale, indicating better scalability.

(ii) **Context efficiency via trajectory compression in *Compressed Reasoning Aggregation*.** As shown

in Figure 5 (*Right*), compressing the agentic reasoning trajectory reduces context token usage by up to **99%** relative to the full trajectory, achieving an almost complete compression. This enables multi-trajectory reasoning aggregation within context limits and improves processing efficiency.

In summary, the proposed PARALLELMUSE is not a conventional test-time scaling strategy that improves performance by aggressively sacrificing efficiency. Instead, by leveraging a task-informed design, it scales computation where it matters most, allocating additional tokens and reasoning capacity as needed to high-utility regions while eliminating most redundant or avoidable computation.

4.6 Impact of Model Capability on Compressed Reasoning Aggregation

In the proposed PARALLELMUSE, the compression process in *Compressed Reasoning Aggregation* can be viewed as extracting and reconstructing the agent's internal information state graph $\mathcal{G}$ (defined in equation 2) from the full agentic reasoning trajectory. This graph, described by the compressed report, encapsulates all information necessary for answer derivation. Hence, the quality of compression depends on the fidelity of this extraction and reconstruction, which directly affects subsequent aggregation performance.

Table 4: Performance gains from using stronger models for *Compressed Reasoning Aggregation*. Rollout configuration detailed in Table 1.

Rollout Model	Aggregation Model	BrowseComp
GPT-OSS-20B	GPT-OSS-20B	49.0
	GPT-OSS-120B ↑	50.5
	GPT-5 ↑↑	**55.5**
Tongyi-DR-30B-A3B	Tongyi-DR-30B-A3B	65.0
	GPT-5 ↑	**66.0**

To examine whether a stronger model can perform higher-quality compression and yield better aggregation, we evaluate the setting where the *Compressed Reasoning Aggregation* stage is executed by models stronger than those used for the first-stage partial rollout. As shown in Table 4, when the first-stage sampling is conducted with GPT-OSS-20B, replacing it with a stronger GPT-OSS-120B for the aggregation stage leads to a clear performance improvement. Further substitution with GPT-5 brings continuous gains, and a similar trend is observed on the Tongyi-DR-30B-A3B model, confirming that the compressed report effectively represents the agent's internal information state graph and that higher-quality graph reconstruction enhances overall performance. This result also suggests a practical insight for multi-agent design (Han et al., 2024; Li et al., 2024): combining models of different strengths can balance efficiency and performance.

5 Related Work

5.1 Deep Information-Seeking Agents

Deep information-seeking (IS) agents are autonomous systems designed to engage in multi-step interaction with external information environments (such as the web) and integrate retrieved data through reasoning in order to address complex knowledge-intensive tasks.

The development of such agents has benefited from both proprietary and open-source initiatives. On the proprietary side, systems from major research organizations have set benchmarks for deep exploration and reasoning capabilities (OpenAI, 2025b; Team, 2025a; AI, 2025; Perplexity, 2025; Anthropic, 2025; DeepMind, 2025), though their architectures and training protocols remain closed. On the open-source front, community efforts have advanced transparency and reproducibility in deep IS agent design (Zhang et al., 2025a; Wu et al., 2025c;a; Li et al., 2025c;b; Tao et al., 2025; Geng et al., 2025; Sun et al., 2025; Gao et al., 2025; Liu et al., 2025; Lu et al., 2025; Team, 2025b), driving continuous progress in this domain.

In this work, we further explore the unique characteristics of deep IS agents and propose PARALLELMUSE to exploit these properties more effectively, thereby enhancing both capability and efficiency.

5.2 Parallel Thinking for Test-Time Scaling

Parallel thinking (Wang et al., 2025) serves as an effective test-time scaling strategy for reasoning, particularly in agentic settings that require deep and complex interactions. It generates multiple reasoning trajectories to capture diverse reasoning behaviors and jointly determines the final answer.

Conceptually, parallel thinking follows a two-stage paradigm (Li et al., 2025a): *exploratory sampling* and *answer generation*. The first stage explores diverse reasoning paths through independent sampling (Wei et al., 2022; Zeng et al., 2025), structured rollouts, or intermediate partial rollouts (AI, 2025; Dong et al., 2025; Hou et al., 2025; Li et al., 2025e) where branches share context but remain flexible. In agentic reasoning with vast exploration spaces, independent and partial rollouts are generally more effective and efficient than structured ones. The second stage focuses on synthesizing results via either answer selection (Wang et al., 2022; Fu et al., 2025), which is efficient but often biased, or answer aggregation (Jiang et al., 2023; Liang et al., 2024; Zhang et al., 2025b; Qiao et al., 2025), which is more stable but challenged by the need to identify which intermediate reasoning contributes most to the final outcome.

Most existing parallel thinking methods inherit the assumptions of pure reasoning tasks. Building on a detailed analysis of agentic reasoning, especially in deep IS tasks, we propose PARALLELMUSE, a paradigm that fully leverages these properties to more effectively unlock the potential of deep IS agents.

6 Conclusion

This work investigates the challenges of applying parallel thinking to deep information-seeking (IS) agents. Conventional parallel thinking strategies often waste computation through redundant rollouts and struggle to integrate long-horizon reasoning due to limited context capacity. Building on an in-depth analysis of the characteristics of deep IS tasks, we incorporate these insights into method design and propose PARALLELMUSE, a two-stage paradigm that enhances both exploration efficiency and reasoning aggregation. Experimental results across multiple open-source agents and benchmarks demonstrate that PARALLELMUSE achieves substantial performance improvements while greatly reducing exploratory token consumption, highlighting its effectiveness for efficient and scalable deep IS reasoning.

7 Limitations and Future Work

In this work, we focus primarily on question-answering–oriented deep IS tasks, where the toolset is limited to Search and Visit. While this configuration is optimal for deep IS tasks, more general agentic tasks often involve a broader range of tools (Fang et al., 2025), leading to substantially larger exploration spaces. Designing effective parallel thinking strategies under such complex tool configurations to extend applicability to general agentic settings remains an open direction for future research.

References

Skywork AI. Skywork-deepresearch. https://github.com/SkyworkAI/Skywork-DeepResearch, 2025.

Gene M Amdahl. Validity of the single processor approach to achieving large scale computing capabilities. In *Proceedings of the April 18-20, 1967, spring joint computer conference*, pp. 483–485, 1967.

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

Cameron B Browne, Edward Powley, Daniel Whitehouse, Simon M Lucas, Peter I Cowling, Philipp Rohlfshagen, Stephen Tavener, Diego Perez, Spyridon Samothrakis, and Simon Colton. A survey of monte carlo tree search methods. *IEEE Transactions on Computational Intelligence and AI in games*, 4(1): 1–43, 2012.

Prateek Chhikara. Mind the confidence gap: Overconfidence, calibration, and distractor effects in large language models. *arXiv preprint arXiv:2502.11028*, 2025.

Google DeepMind. Gemini 2.5, 2025. URL https://blog.google/technology/google-deepmind/gemini-model-thinking-updates-march-2025/.

Guanting Dong, Hangyu Mao, Kai Ma, Licheng Bao, Yifei Chen, Zhongyuan Wang, Zhongxia Chen, Jiazhen Du, Huiyang Wang, Fuzheng Zhang, et al. Agentic reinforced policy optimization. *arXiv preprint arXiv:2507.19849*, 2025.

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.

Yichao Fu, Xuewei Wang, Yuandong Tian, and Jiawei Zhao. Deep think with confidence. *arXiv preprint arXiv:2508.15260*, 2025.

Jiaxuan Gao, Wei Fu, Minyang Xie, Shusheng Xu, Chuyi He, Zhiyu Mei, Banghua Zhu, and Yi Wu. Beyond ten turns: Unlocking long-horizon agentic search with large-scale asynchronous rl. *arXiv preprint arXiv:2508.07976*, 2025.

Xinyu Geng, Peng Xia, Zhen Zhang, Xinyu Wang, Qiuchen Wang, Ruixue Ding, Chenxi Wang, Jialong Wu, Yida Zhao, Kuan Li, et al. Webwatcher: Breaking new frontier of vision-language deep research agent. *arXiv preprint arXiv:2508.05748*, 2025.

Jiawei Gu, Xuhui Jiang, Zhichao Shi, Hexiang Tan, Xuehao Zhai, Chengjin Xu, Wei Li, Yinghan Shen, Shengjie Ma, Honghao Liu, et al. A survey on llm-as-a-judge. *arXiv preprint arXiv:2411.15594*, 2024.

Shanshan Han, Qifan Zhang, Yuhang Yao, Weizhao Jin, and Zhaozhuo Xu. Llm multi-agent systems: Challenges and open problems. *arXiv preprint arXiv:2402.03578*, 2024.

Zhenyu Hou, Ziniu Hu, Yujiang Li, Rui Lu, Jie Tang, and Yuxiao Dong. Treerl: Llm reinforcement learning with on-policy tree search. *arXiv preprint arXiv:2506.11902*, 2025.

Chaeyun Jang, Hyungi Lee, Seanie Lee, and Juho Lee. Calibrated decision-making through llm-assisted retrieval. *arXiv preprint arXiv:2411.08891*, 2024.

Dongfu Jiang, Xiang Ren, and Bill Yuchen Lin. Llm-blender: Ensembling large language models with pairwise ranking and generative fusion. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 14165–14178, 2023.

Baixuan Li, Yunlong Fan, Tianyi Ma, Miao Gao, Chuanqi Shi, and Zhiqiang Gao. Raspberry: Retrieval-augmented monte carlo tree self-play with reasoning consistency for multi-hop question answering. In *Findings of the Association for Computational Linguistics: ACL 2025*, pp. 11258–11276, 2025a.

Haoyang Li, Yiming Li, Anxin Tian, Tianhao Tang, Zhanchao Xu, Xuejia Chen, Nicole Hu, Wei Dong, Qing Li, and Lei Chen. A survey on large language model acceleration based on kv cache management. *arXiv preprint arXiv:2412.19442*, 2024.

Kuan Li, Zhongwang Zhang, Huifeng Yin, Rui Ye, Yida Zhao, Liwen Zhang, Litu Ou, Dingchu Zhang, Xixi Wu, Jialong Wu, Xinyu Wang, Zile Qiao, Zhen Zhang, Yong Jiang, Pengjun Xie, Fei Huang, and Jingren Zhou. Websailor-v2: Bridging the chasm to proprietary agents via synthetic data and scalable reinforcement learning, 2025b. URL https://arxiv.org/abs/2509.13305.

Kuan Li, Zhongwang Zhang, Huifeng Yin, Liwen Zhang, Litu Ou, Jialong Wu, Wenbiao Yin, Baixuan Li, Zhengwei Tao, Xinyu Wang, Weizhou Shen, Junkai Zhang, Dingchu Zhang, Xixi Wu, Yong Jiang, Ming Yan, Pengjun Xie, Fei Huang, and Jingren Zhou. Websailor: Navigating super-human reasoning for web agent, 2025c. URL https://arxiv.org/abs/2507.02592.

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.

Yizhi Li, Qingshui Gu, Zhoufutu Wen, Ziniu Li, Tianshun Xing, Shuyue Guo, Tianyu Zheng, Xin Zhou, Xingwei Qu, Wangchunshu Zhou, et al. Treepo: Bridging the gap of policy optimization and efficacy and inference efficiency with heuristic tree-based modeling. *arXiv preprint arXiv:2508.17445*, 2025e.

Tian Liang, Zhiwei He, Wenxiang Jiao, Xing Wang, Yan Wang, Rui Wang, Yujiu Yang, Shuming Shi, and Zhaopeng Tu. Encouraging divergent thinking in large language models through multi-agent debate. In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, pp. 17889–17904, 2024.

Aixin Liu, Bei Feng, Bing Xue, Bingxuan Wang, Bochao Wu, Chengda Lu, Chenggang Zhao, Chengqi Deng, Chenyu Zhang, Chong Ruan, et al. DeepSeek-V3 technical report. *arXiv preprint arXiv:2412.19437*, 2024.

Junteng Liu, Yunji Li, Chi Zhang, Jingyang Li, Aili Chen, Ke Ji, Weiyu Cheng, Zijia Wu, Chengyu Du, Qidi Xu, et al. Webexplorer: Explore and evolve for training long-horizon web agents. *arXiv preprint arXiv:2509.06501*, 2025.

Rui Lu, Zhenyu Hou, Zihan Wang, Hanchen Zhang, Xiao Liu, Yujiang Li, Shi Feng, Jie Tang, and Yuxiao Dong. Deepdive: Advancing deep search agents with knowledge graphs and multi-turn rl. *arXiv preprint arXiv:2509.10446*, 2025.

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. gpt-oss-120b & gpt-oss-20b model card, 2025a. URL https://arxiv.org/abs/2508.10925.

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

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

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, Shen Huang, Jialong Wu, Kuan Li, Wenbiao Yin, Xinyu Wang, Liwen Zhang, Baixuan Li, Zhengwei Tao, Weizhou Shen, Xixi Wu, Yong Jiang, Pengjun Xie, Fei Huang, Jun Zhang, and Jingren Zhou. WebResearcher: Unleashing unbounded reasoning capability in long-horizon agents, 2025.

Liangcai Su, Zhen Zhang, Guangyu Li, Zhuo Chen, Chenxi Wang, Maojia Song, Xinyu Wang, Kuan Li, Jialong Wu, Xuanzhong Chen, et al. Scaling agents via continual pre-training. *arXiv preprint arXiv:2509.13310*, 2025.

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

Zhengwei Tao, Jialong Wu, Wenbiao Yin, Junkai Zhang, Baixuan Li, Haiyang Shen, Kuan Li, Liwen Zhang, Xinyu Wang, Yong Jiang, Pengjun Xie, Fei Huang, and Jingren Zhou. WebShaper: Agentically data synthesizing via information-seeking formalization, 2025.

Kimi Team. Kimi researcher tech report, 2025a. URL https://moonshotai.github.io/Kimi-Researcher/.

Tongyi DeepResearch Team. Tongyi deepresearch: A new era of open-source ai researchers. https://github.com/Alibaba-NLP/DeepResearch, 2025b.

Xuezhi Wang, Jason Wei, Dale Schuurmans, Quoc Le, Ed Huai hsin Chi, and Denny Zhou. Self-consistency improves chain of thought reasoning in language models. *ArXiv*, abs/2203.11171, 2022.

Ziqi Wang, Boye Niu, Zipeng Gao, Zhi Zheng, Tong Xu, Linghui Meng, Zhongli Li, Jing Liu, Yilong Chen, Chen Zhu, et al. A survey on parallel reasoning. *arXiv preprint arXiv:2510.12164*, 2025.

Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, Fei Xia, Ed Chi, Quoc V Le, Denny Zhou, et al. Chain-of-thought prompting elicits reasoning in large language models. *Advances in neural information processing systems*, 35:24824–24837, 2022.

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, Gang Fu, Yong Jiang, Pengjun Xie, Fei Huang, and Jingren Zhou. Webdancer: Towards autonomous information seeking agency, 2025a. URL https://arxiv.org/abs/2505.22648.

Jialong Wu, Zhenglin Wang, Linhai Zhang, Yilong Lai, Yulan He, and Deyu Zhou. SCOPE: Optimizing key-value cache compression in long-context generation. In Wanxiang Che, Joyce Nabende, Ekaterina Shutova, and Mohammad Taher Pilehvar (eds.), *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 10775–10790, Vienna, Austria, July 2025b. Association for Computational Linguistics. ISBN 979-8-89176-251-0. doi: 10.18653/v1/2025.acl-long.529. URL https://aclanthology.org/2025.acl-long.529/.

Jialong Wu, Wenbiao Yin, Yong Jiang, Zhenglin Wang, Zekun Xi, Runnan Fang, Linhai Zhang, Yulan He, Deyu Zhou, Pengjun Xie, and Fei Huang. Webwalker: Benchmarking llms in web traversal, 2025c. URL https://arxiv.org/abs/2501.07572.

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.

Weihao Zeng, Keqing He, Chuqiao Kuang, Xiaoguang Li, and Junxian He. Pushing test-time scaling limits of deep search with asymmetric verification. *arXiv preprint arXiv:2510.06135*, 2025.

Dingchu Zhang, Yida Zhao, Jialong Wu, Baixuan Li, Wenbiao Yin, Liwen Zhang, Yong Jiang, Yufeng Li, Kewei Tu, Pengjun Xie, and Fei Huang. Evolvesearch: An iterative self-evolving search agent, 2025a. URL https://arxiv.org/abs/2505.22501.

Qiyuan Zhang, Fuyuan Lyu, Zexu Sun, Lei Wang, Weixu Zhang, Zhihan Guo, Yufei Wang, Irwin King, Xue Liu, and Chen Ma. What, how, where, and how well? a survey on test-time scaling in large language models. *CoRR*, 2025b.

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.