Agentic Parallel Thinking for Deep Information Seeking

Anonymous ACL submission

Abstract

Parallel thinking broadens exploration and complements deep information-seeking (IS) agents, but in this setting it is hindered by redundant from-scratch rollouts and context-limited answer generation that cannot reliably integrate long-horizon trajectories. To address these issues, we propose PARALLELMUSE, a two-stage inference-only 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, 2025a; Team, 2025a,b) uncover hard-to-access information by iterating between environmental[2] interaction and internal deliberation, progressively deepening reasoning during a single execution to solve complex problems (Wu et al., 2025a; Li et al., 2025c; Tao et al., 2025), extending LLMs beyond static training data to reason over real-world knowledge. In this setting, parallel thinking provides a natural form of test-time scaling: increasing the number of parallel exploration paths broadens search while preserving reasoning depth along each path, improving performance without updating model parameters. Consistent with the two-stage view of parallel thinking (Li et al., 2025a) (exploratory sampling and answer generation), we adapt it to deep IS agents in PARALLELMUSE and characterize the two stages under agentic interaction and propose stage-wise optimizations.

First, in the exploratory sampling stage, conventional rollout strategies in parallel thinking typically restart from scratch at each iteration, resampling the entire exploration space (Fu et al., 2025; Zeng et al., 2025). During certain reasoning phases, however, exploration diversity is inherently low, making repeated rollouts inefficient. 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 must distill one output from multiple candidates via *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). In

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

[2]The term "environment" specifically refers to the web or information sources with which the IS agent interacts.

complex agentic tasks, correct outcomes can be sparse in a vast sampling space, and continual incorporation of external, non–model-generated information shifts the output distribution and undermines confidence calibration (Jang et al., 2024), so majority voting (Wang et al., 2022) and confidence-based selection (Fu et al., 2025) often fail. Aggregation then trades off between ignoring intermediate reasoning (final-answer only) and exceeding context limits (full-trace), while aggregating only the last steps (Qiao et al., 2025) shortens sequences at the cost of discarding earlier planning and decomposition cues crucial for coherence judgments.

To address this challenge, we view IS as discovering key entities and constructing connections among them (Li et al., 2025c; Tao et al., 2025). Pilot observations suggest that only a small subset of explored entities ultimately contributes to the answer, indicating substantial redundancy and strong potential for lossless compression of trajectories. Accordingly, we propose ***Compressed Reasoning Aggregation*** as the second stage of PARALLELMUSE: it condenses each candidate trajectory into a concise, answer-relevant report, then aggregates these reports to produce the final output. This enables joint consideration of multiple trajectories while reducing reliance on majority-based selection, yielding more reliable answer generation.

Our PARALLELMUSE is evaluated on four open-source deep IS agents, across four challenging benchmarks that jointly assess deep search and reasoning abilities. 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.

2 Pilot Observation

We begin with a preliminary analysis of the characteristics of deep IS agents and tasks, providing insights from two perspectives: (i) exploratory sampling and (ii) the resulting 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 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, $\mathcal{T}^e = \emptyset$, 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.

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

where $x_{t,i} \in \mathcal{T}_t^f$ and $f \in \{r, e\}$ represents the functional region, which is partitioned into a reasoning region r and an exploration region e.

We analyze $\mathrm{PPL}(r,t)$ and $\mathrm{PPL}(e,t)$ 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* is highest at the beginning of a task, when the agent has not yet acquired external information and must explore the environment with minimal prior knowledge. *Reasoning uncertainty* peaks slightly later in the early stage, as the agent starts to integrate newly retrieved

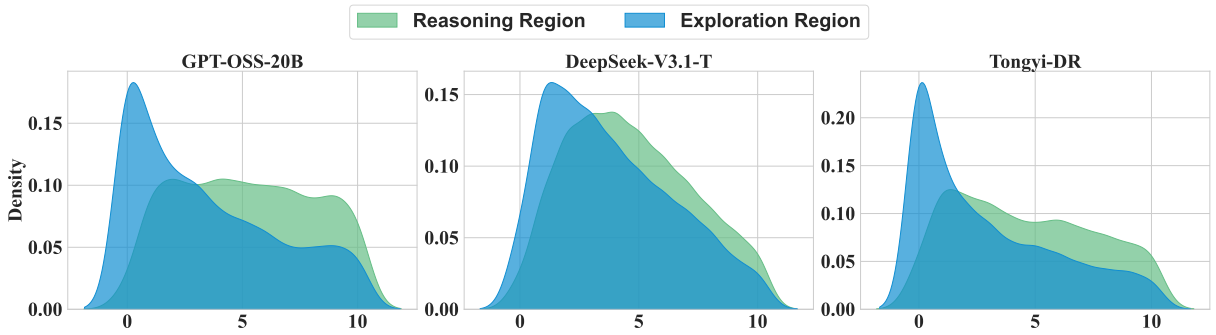

Figure 1: KDE-smoothed distribution of steps with top-4 uncertainty on the BrowseComp subset (truncated to earlier steps as later ones are typically more certain). DeepSeek-V3.1-T denotes DeepSeek-V3.1-Terminus, and Tongyi-DR denotes Tongyi-DeepResearch-30B-A3B.

information into its reasoning process. As execution continues, both types of uncertainty gradually decrease as knowledge accumulates and reasoning becomes more grounded, leading to increasingly confident decisions and actions.

This observation further informs the design of the ***Functionality-Specified Partial Rollout*** 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* relevant for answer derivation, and $\mathcal{R}_t \subseteq \mathcal{V}_t \times \mathcal{V}_t$ represents the relations among them. The task objective is to iteratively refine the graph $\mathcal{G}_t$ until it captures the entities and relations necessary for deriving the final answer. Upon termination, the resulting graph $\mathcal{G}_{\text{final}} \supseteq \mathcal{I}_{\text{answer}}$ encodes all information essential for answer derivation and serves as the core representation of the reasoning trajectory.

Based on this formulation of deep IS tasks, we estimate trajectory redundancy by the proportion

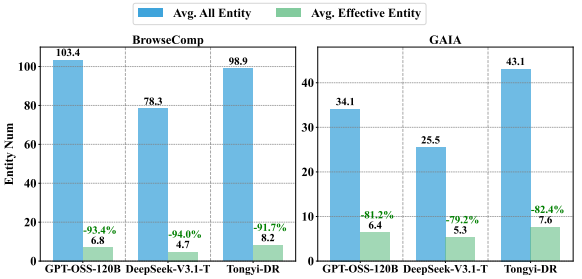

Figure 2: Average entity count per task and per model, where entities are extracted by GPT-4.1 based on the complete reasoning trajectory and ground-truth answer.

of effective entities, namely those that directly or indirectly contribute to answer derivation, among all entities discovered during execution. Formally, let $\mathcal{V}_{\text{total}}$ denote the set of all entities explored by the agent, and let $\mathcal{V}_{\text{eff}} \subseteq \mathcal{V}_{\text{total}}$ denote the subset that is useful for deriving the final answer. We define the redundancy ratio Γred 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 trajectory, corresponding to a stronger potential for lossless compression. Here, lossless compression refers to eliminating redundant entities and reasoning steps while preserving all information required for complete answer derivation, namely the reconstruction of $\mathcal{G}_{\text{final}}$. Accordingly, Γ_{red} serves as an approximate indicator 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

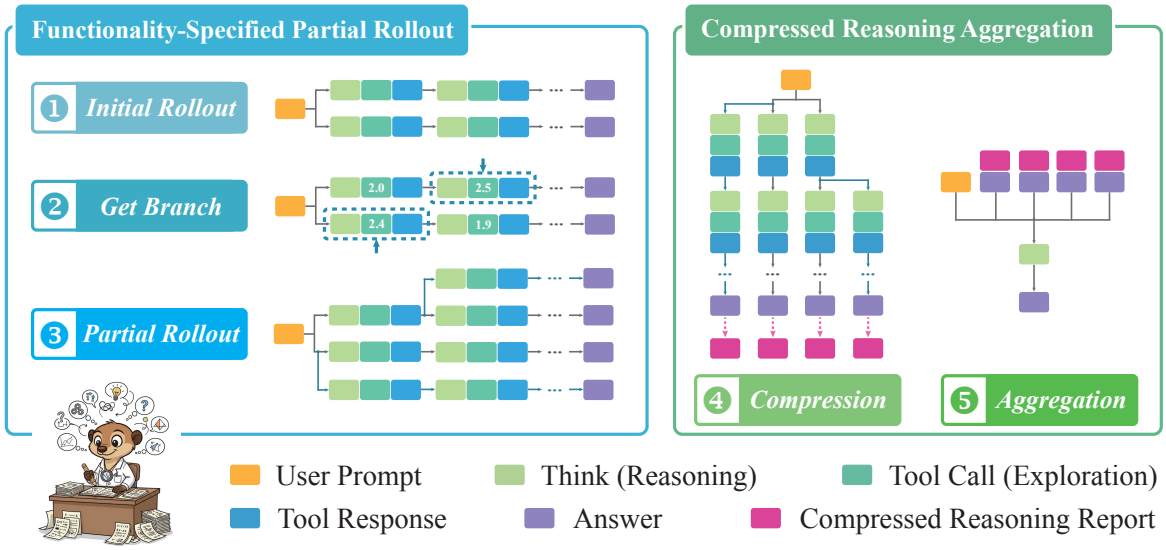

Figure 3: Workflow of PARALLELMUSE, including (*Left*) the Functionality-Specified Partial Rollout, where the *Get Branch* shows the selection of top-k steps based on (exploration) tool-call uncertainty (just as an example of branching criterion), and (*Right*) the Compressed Reasoning Aggregation.

Compressed Reasoning Aggregation method in PARALLELMUSE, which aims to integrate as much effective reasoning information as possible into final aggregation with minimal information loss.

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

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 Eq. (1), to quantify the 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 Eq. (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 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 for parallelization while preserving each branch's causal decoding consistency, enabling multiple branches to expand concurrently. 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 Eq. (4) and Eq. (5) yields the theoretical overall speedup: $\text{Speedup}_{\text{total}} \lesssim \text{ReuseFactor} \times \text{ParaFactor}(P)$. In practice with 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 analysis in Section 2.2, we observe that reasoning trajectories produced during exploratory sampling in deep IS tasks are highly redundant and thus largely compressible. To incorporate rich intermediate reasoning signals into answer aggregation while avoiding context overflow and excessive computation, we compress each exploratory reasoning trajectory prior to aggregation.

As illustrated in Figure 3 (*Right*), each trajectory is distilled into a structured report that preserves information essential for answer derivation. Specifically, the report records: (i) solution planning, describing the decomposition of the problem into subproblems and their execution order; (ii) solution methods, detailing the tools used, and intermediate results; and (iii) final reasoning, explaining how subproblem solutions are integrated to obtain the final answer. Redundant tool responses and unproductive reasoning steps are discarded. This compression effectively reconstructs the agent's internal information-state graph $\mathcal{G}$ (Eq. (2)), capturing all information relevant to answer derivation.

Reasoning-Guided Answer Aggregation. After compression, we jointly aggregate the N reports within a single context window, enabling holistic evaluation of the compressed reasoning trajectories rather than relying solely on final answers or partial traces. This supports more reliable answer selection based on reasoning coherence. During aggregation, we explicitly discourage majority bias and prohibit trivial enumeration of answers. Each report already contains sufficient tool-calling provenance for answer derivation. Accordingly, the aggregation stage performs no additional tool invocation and reasons solely over the N reports, yielding an effective yet computationally efficient process.

4 Experiments

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). Together, these benchmarks cover complementary aspects of deep IS, with BrowseComp and BrowseComp-zh emphasizing deep search, HLE focusing on reasoning, and GAIA balancing both.

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.

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

Baselines. In addition to the standard inference baseline without parallel thinking (*No Scaling*), we compare PARALLELMUSE with several representative parallel thinking baselines: **(i)** Self-Consistency (*Majority Vote*) (Wang et al., 2022), which selects the most frequent answer across multiple trajectories; **(ii)** *Max #Tool Call* (Zeng et al., 2025), which selects the answer from the trajectory with the largest number of tool calls;

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‡
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
	Answer Aggr.	38.0	36.0	61.2	26.8
	Reasoning Aggr.	24.5	18.0	61.2	27.4
	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
	Answer Aggr.	46.5	46.7	76.7	37.6
	Reasoning Aggr.	27.5	24.6	72.8	39.5
	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
	Answer Aggr.	31.5	40.1	70.9	33.1
	Reasoning Aggr.	30.5	32.9	72.8	29.3
	PARALLELMUSE	**39.0**	**50.2**	**74.8**	**37.6**
Tongyi-DR	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
	Answer Aggr.	62.5	54.3	78.6	45.9
	Reasoning Aggr.	61.5	55.0	77.7	45.2
	PARALLELMUSE	**65.0**	**57.1**	**79.6**	**52.2**

Table 1: Overall performance. Scores marked with ‡ represent full-benchmark results, whereas unmarked scores correspond to our benchmark settings. Evaluation details can be found in Appendix A.1.

(iii) DeepConf (*Weighted Vote*) (Fu et al., 2025), which weights candidate answers by the model's self-estimated confidence; **(iv)** Generative Self-Aggregation (*Answer Aggr.*) (Li et al., 2025f), which aggregates only final answers; and **(v)** Recursive Self-Aggregation (*Reasoning Aggr.*) (Venkatraman et al., 2025), which aggregates sampled sub-trajectories from each trajectory.

4.2 Overall Performance

We report results of closed-source deep IS agents and compare them with open-source agents augmented with PARALLELMUSE and representative parallel thinking baselines. As shown in Table 1, PARALLELMUSE consistently delivers the largest performance gains across all agent models and benchmarks. When applied to Tongyi-DR, it achieves performance comparable to or even exceeding that of most closed-source agents.

We further observe that *Weighted Vote*, which relies on self-estimated confidence, underperforms *Majority Vote* for all models except Tongyi-DR. This is attributable to confidence miscalibration in agentic settings: repeated incorporation of external, non-parametric content (e.g., tool responses) shifts the model's internal distributions, degrading confidence reliability (Jang et al., 2024; Chhikara, 2025). Exceptions include the HLE benchmark, which involves limited external interaction, and Tongyi-DR, whose continual pre-training improves calibration

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

Table 2: Performance comparison between full from-scratch rollouts and partial rollouts guided by functional-region uncertainty. Detailed configurations are listed in Appendix A.1.

over tool responses (Su et al., 2025). In contrast, PARALLELMUSE avoids confidence-based selection and aggregates near-lossless agentic reasoning signals, leading to consistent and substantial improvements across all baselines.

4.3 Analysis of Partial Rollout over Distinct Functional Regions

To analyze the impact of functional-region–aware branching in partial rollout, Table 2 reports the average pass rate over 8 rollouts. *From Scratch* denotes full rollouts without context reuse. For PARALLELMUSE, we compare three strategies: branching based on uncertainty from the *Reasoning* region, the *Exploration* region, and a *Mixed* strategy that draws branching steps evenly from both.

The results show that the effectiveness of region-specific branching varies across models, reflecting differences in their behaviors. For instance, GPT-OSS-120B benefits less from reasoning-based branching due to its strong adaptive reasoning, whereas DeepSeek-V3.1-T shows greater gains from reasoning uncertainty, as its function calling occurs outside the `thinking` mode. These findings inform our design choices in PARALLELMUSE.

Across most settings, partial rollout consistently outperforms full rollout by enabling more targeted exploration. In deep IS tasks, uncertainty-guided branching allocates sampling budget to high-uncertainty steps, analogous to Monte-Carlo Tree Search (MCTS) (Browne et al., 2012), thereby avoiding local optima and improving efficiency.

4.4 Performance Gains from Compressed Reasoning Aggregation

In this section, we isolate the effectiveness of our second-stage answer aggregation method. As shown in Figure 4, even without exploration gains

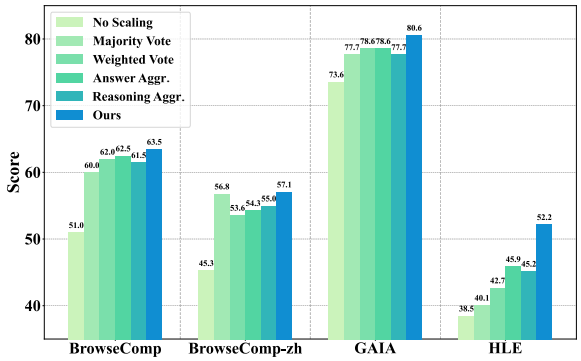

Figure 4: Performance gains from different answer generation methods, with sampling fixed to 8 from-scratch rollouts to isolate sampling (exploration) effects.

from first-stage partial rollout, *Compressed Reasoning Aggregation* alone delivers the largest performance improvement. The method performs near-lossless compression of each agentic reasoning trajectory, enabling efficient integration of rich reasoning information without additional tool calls. By fully exploiting the sampled information during aggregation, it improves solution quality while maintaining high efficiency.

4.5 Efficiency Gains through Context Reuse and Trajectory Compression

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

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

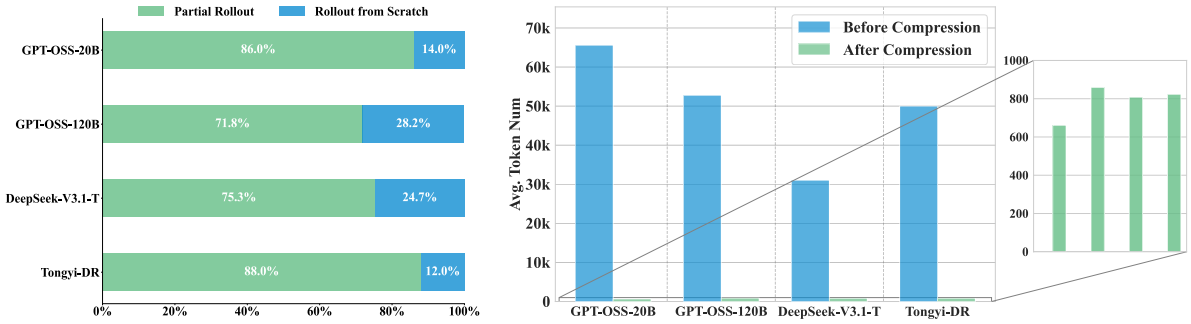

Figure 5: Efficiency gains using PARALLELMUSE. (i) (*Left*) Token reduction through context reuse in our partial rollout method. We use token consumption per trajectory under from-scratch rollouts as the baseline. Green bars denote token cost with partial rollout (numbers indicate the baseline ratio), and blue bars indicate tokens saved. (ii) (*Right*) Comparison of context token usage before and after trajectory compression.

Context Efficiency via Trajectory Compression. As shown in Figure 5 (*Right*), compressing the agent 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, PARALLELMUSE is not a conventional test-time scaling strategy that trades efficiency for performance. Instead, it scales computation selectively, allocating additional tokens to high-utility regions while eliminating most redundant and avoidable computation.

5 Related Work

5.1 Deep Information-Seeking Agents

Deep information-seeking (IS) agents are autonomous systems that interact with external information environments, primarily the web, through multi-step exploration and reasoning to solve complex knowledge-intensive tasks. Progress in this area has been driven by both proprietary and open-source efforts. Proprietary systems demonstrate strong deep exploration and reasoning capabilities (OpenAI, 2025a; Team, 2025a; AI, 2025; DeepMind, 2025), though their architectures and training pipelines remain opaque. In parallel, open-source initiatives have advanced transparent and reproducible IS agent design (Zhang et al., 2025a; Wu et al., 2025c,a; Li et al., 2025c; Tao et al., 2025; Liu et al., 2025; Lu et al., 2025; Team, 2025b), enabling steady community-driven progress.

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

5.2 Parallel Thinking for Test-Time Scaling

Parallel thinking (Wang et al., 2025) is a test-time scaling strategy that improves reasoning by generating multiple trajectories to capture diverse behaviors and jointly selecting a 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), partial rollouts with shared context (AI, 2025), or structured rollouts (Qi et al., 2025), with independent and partial rollouts generally more effective in large agentic search spaces. The second stage synthesizes results through 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), which is more stable but faces challenges in identifying which intermediate reasoning is most critical to the final answer.

Most existing parallel thinking methods inherit the assumptions of pure reasoning tasks. Building on an analysis of agentic reasoning in deep IS settings, we propose PARALLELMUSE, which leverages these properties to more effectively unlock the potential of deep IS agents.

6 Conclusion

This work studies the limitations of parallel thinking in deep IS agents and introduces PARALLELMUSE, a two-stage paradigm that improves exploration efficiency and reasoning aggregation. By leveraging the characteristics of deep IS tasks, PARALLELMUSE delivers substantial performance gains across multiple agents while significantly reducing exploratory token consumption.

Limitations and Future Work

In this work, we focus primarily on QA–oriented deep IS tasks, where the toolset is limited to Search and Visit. While this configuration is sufficient 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. 2025. Skywork-deepresearch. https://github.com/SkyworkAI/Skywork-DeepResearch.

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

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

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

Google DeepMind. 2025. Gemini 2.5.

Guanting Dong, Hangyu Mao, Kai Ma, Licheng Bao, Yifei Chen, Zhongyuan Wang, Zhongxia Chen, Jiazhen Du, Huiyang Wang, Fuzheng Zhang, and 1 others. 2025. Agentic reinforced policy optimization. *arXiv preprint arXiv:2507.19849*.

Runnan Fang, Shihao Cai, Baixuan Li, Jialong Wu, Guangyu Li, Wenbiao Yin, Xinyu Wang, Xiaobin Wang, Liangcai Su, Zhen Zhang, and 1 others. 2025. Towards general agentic intelligence via environment scaling. *arXiv preprint arXiv:2509.13311*.

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

Jiawei Gu, Xuhui Jiang, Zhichao Shi, Hexiang Tan, Xuehao Zhai, Chengjin Xu, Wei Li, Yinghan Shen, Shengjie Ma, Honghao Liu, and 1 others. 2024. A survey on llm-as-a-judge. *arXiv preprint arXiv:2411.15594*.

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

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

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

Dongfu Jiang, Xiang Ren, and Bill Yuchen Lin. 2023. 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)*, pages 14165–14178.

Baixuan Li, Yunlong Fan, Tianyi Ma, Miao Gao, Chuanqi Shi, and Zhiqiang Gao. 2025a. 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*, pages 11258–11276.

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

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. 2025b. Websailor-v2: Bridging the chasm to proprietary agents via synthetic data and scalable reinforcement learning. *Preprint*, arXiv: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. 2025c. Websailor: Navigating super-human reasoning for web agent. *Preprint*, arXiv:2507.02592.

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

Yizhi Li, Qingshui Gu, Zhoufutu Wen, Ziniu Li, Tianshun Xing, Shuyue Guo, Tianyu Zheng, Xin Zhou, Xingwei Qu, Wangchunshu Zhou, and 1 others. 2025e. Treepo: Bridging the gap of policy optimization and efficacy and inference efficiency with heuristic tree-based modeling. *arXiv preprint arXiv:2508.17445*.

Zichong Li, Xinyu Feng, Yuheng Cai, Zixuan Zhang, Tianyi Liu, Chen Liang, Weizhu Chen, Haoyu Wang, and Tuo Zhao. 2025f. Llms can generate a better answer by aggregating their own responses. *arXiv preprint arXiv:2503.04104*.

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

Aixin Liu, Bei Feng, Bing Xue, Bingxuan Wang, Bochao Wu, Chengda Lu, Chenggang Zhao, Chengqi Deng, Chenyu Zhang, Chong Ruan, and 1 others. 2024. DeepSeek-V3 technical report. *arXiv preprint arXiv:2412.19437*.

Junteng Liu, Yunji Li, Chi Zhang, Jingyang Li, Aili Chen, Ke Ji, Weiyu Cheng, Zijia Wu, Chengyu Du, Qidi Xu, and 1 others. 2025. Webexplorer: Explore and evolve for training long-horizon web agents. *arXiv preprint arXiv:2509.06501*.

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

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

OpenAI. 2025a. Deep research system card.

OpenAI. 2025b. gpt-oss-120b & gpt-oss-20b model card. *Preprint*, arXiv:2508.10925.

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

Zhenting Qi, MA Mingyuan, Jiahang Xu, Li Lyna Zhang, Fan Yang, and Mao Yang. 2025. Mutual reasoning makes smaller llms stronger problem-solver. In *The Thirteenth International Conference on Learning Representations*.

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. 2025. WebResearcher: Unleashing unbounded reasoning capability in long-horizon agents. *Preprint*, arXiv:2509.13309.

Liangcai Su, Zhen Zhang, Guangyu Li, Zhuo Chen, Chenxi Wang, Maojia Song, Xinyu Wang, Kuan Li, Jialong Wu, Xuanzhong Chen, and 1 others. 2025. Scaling agents via continual pre-training. *arXiv preprint arXiv:2509.13310*.

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. 2025. WebShaper: Agentically data synthesizing via information-seeking formalization. *Preprint*, arXiv:2507.15061.

Kimi Team. 2025a. Kimi researcher tech report.

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

Siddarth Venkatraman, Vineet Jain, Sarthak Mittal, Vedant Shah, Johan Obando-Ceron, Yoshua Bengio, Brian R Bartoldson, Bhavya Kailkhura, Guillaume Lajoie, Glen Berseth, and 1 others. 2025. Recursive self-aggregation unlocks deep thinking in large language models. *arXiv preprint arXiv:2509.26626*.

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

Ziqi Wang, Boye Niu, Zipeng Gao, Zhi Zheng, Tong Xu, Linghui Meng, Zhongli Li, Jing Liu, Yilong Chen, Chen Zhu, and 1 others. 2025. A survey on parallel reasoning. *arXiv preprint arXiv:2510.12164*.

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

Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, Fei Xia, Ed Chi, Quoc V Le, Denny Zhou, and 1 others. 2022. Chain-of-thought prompting elicits reasoning in large language models. *Advances in neural information processing systems*, 35:24824–24837.

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. 2025a. Webdancer: Towards autonomous information seeking agency. *Preprint*, arXiv:2505.22648.

Jialong Wu, Zhenglin Wang, Linhai Zhang, Yilong Lai, Yulan He, and Deyu Zhou. 2025b. SCOPE: Optimizing key-value cache compression in long-context generation. In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 10775–10790, Vienna, Austria. Association for Computational Linguistics.

Jialong Wu, Wenbiao Yin, Yong Jiang, Zhenglin Wang, Zekun Xi, Runnan Fang, Linhai Zhang, Yulan He, Deyu Zhou, Pengjun Xie, and Fei Huang. 2025c. Webwalker: Benchmarking llms in web traversal. *Preprint*, arXiv:2501.07572.

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

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

Dingchu Zhang, Yida Zhao, Jialong Wu, Baixuan Li, Wenbiao Yin, Liwen Zhang, Yong Jiang, Yufeng Li, Kewei Tu, Pengjun Xie, and Fei Huang. 2025a. Evolvesearch: An iterative self-evolving search agent. *Preprint*, arXiv:2505.22501.

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

Peilin Zhou, Bruce Leon, Xiang Ying, Can Zhang, Yifan Shao, Qichen Ye, Dading Chong, Zhiling Jin, Chenxuan Xie, Meng Cao, and 1 others. 2025. Browsecomp-zh: Benchmarking web browsing ability of large language models in chinese. *arXiv preprint arXiv:2504.19314*.

A Appendix

A.1 Implementation Details: 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, the default hyper-parameter settings are listed in Table 3. To ensure fair comparison and reproducibility, all agent–specific hyper-parameters are aligned with their official optimal configurations for tool usage.

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

Table 3: Default settings of PARALLELMUSE.

A.2 Implementation Details: 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 goal-relevant information.

A.3 Analysis of Partial Rollout Hyperparameter Configuration

Setting	BrowseComp	#Tokens
Rollout from Scratch	34.9	56372
Partial ($K = 1, T = 7$)	41.3	40511
Partial ($K = 2, T = 3$)	39.9	41196
Partial ($K = 3, T = 2$)	40.2	40605

Table 4: Exploratory performance and token consumption per rollout under different partial rollout hyperparameter configurations.

We further examined the hyperparameter settings used for partial rollout. As shown in Table 4, with the definitions of K and T provided in Table 3, we evaluated the method using GPT-OSS-120B as the agent model, selecting the tool call segment best suited to the model as the functional region for computing partial rollout branches. The results show that varying the hyperparameters of partial rollout has only marginal effects on both performance and token consumption, which supports the robustness of our proposed approach. Notably, across all hyperparameter configurations, partial rollout consistently delivers substantially better performance than rollout from scratch while requiring significantly fewer tokens, providing strong evidence for the effectiveness of the proposed strategy.

A.4 Impact of Model Capability on Compressed Reasoning Aggregation

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

Table 5: Performance gains from using stronger models for *Compressed Reasoning Aggregation*. Rollout configuration detailed in Table 3.

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 (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.

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 5, when the first-stage sampling is conducted with GPT-OSS-20B, replacing it with a stronger GPT-OSS-120B for the aggregation leads to a clear performance improvement. Further substitution with GPT-5 brings continuous gains, and a similar trend is observed

on the Tongyi-DR 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.