When Deep Research Agents Stagnate: Enhancing Reasoning with Retrieval-Aware Agent Control

Anonymous ACL submission

Abstract

In this paper, we analyze the reasoning trajectories of a variety of DRAs and show that existing agents often suffer from *reasoning stagnation*: the majority of iterations contribute little or no improvement to final performance, while agents lack awareness of their trajectories and are therefore ineffective at adapting their search strategies or determining when to terminate. To address this issue, we introduce a set of unsupervised signals and a Retrieval-Aware Agent Controller (RAAC), which assists the agent in selecting optimal actions at each stage of the research process. RAAC incorporates key information retrieval principles, namely *search novelty* and *information coverage*, resulting in more effective reasoning trajectories that improve overall performance while reducing unnecessary iterations, and consequently cost and latency. Specifically on BROWSECOMP-PLUS and across a large set of DRAs, adding RAAC reduces the number of search calls by an average of 14, significantly improves the best-performing DRA on recall and accuracy, and achieves an accuracy gain of up to 10% (3% on average).

1 Introduction

Retrieval-Augmented Generation (RAG) has become the dominant paradigm for grounding Large Language Model (LLM) outputs in external knowledge (Cuconasu et al., 2024; Mallen et al., 2023; Soudani et al., 2024). Deep Research Agents (DRAs) extend RAG by combining tool use (e.g., search engines) with multi-step reasoning, implemented through either instruction-tuned agents (Press et al., 2023; Yao et al., 2023; Li et al., 2025c) or reinforcement learning-trained agents (Jin et al., 2025; Li et al., 2025b).

Recent work on DRAs has largely focused on component-level improvements, such as enhancing retrieval through re-ranking and query reformulation techniques (Chen et al., 2026b; Meng et al., 2026; Sharifymoghaddam and Lin, 2026), or strengthening reasoning by incorporating additional stages such as outline generation and report writing (Li et al., 2026a, 2025d). However, a critical question remains unexplored: *How effectively do these components work together over long reasoning trajectories to achieve the final objective?*

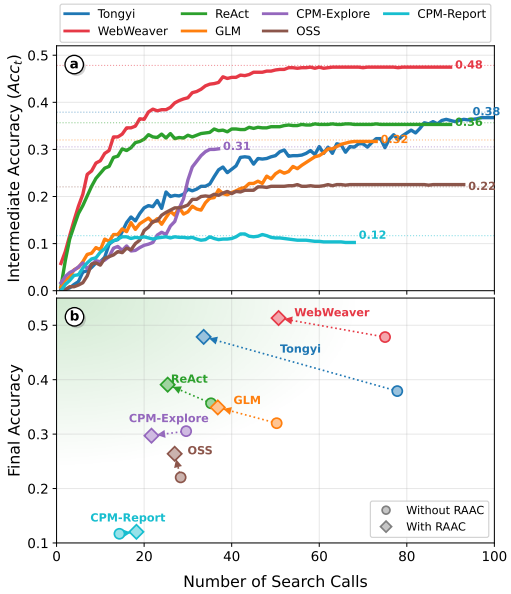

Figure 1: (a) ***Stagnant reasoning*** in DRAs: the agents continue for many unnecessary repetitive steps, even though their information discovery capabilities are saturated and the task often has already been accomplished. (b) Effect of adding RAAC: the introduced unsupervised signals guide DRAs toward a more effective decision-making, improving both efficiency and task success.

In this work, we shift the focus to the intermediate steps of the agent's trajectory and aim to uncover failure patterns in DRAs, irrespective of their underlying design choices. We systematically evaluate seven DRAs with diverse architectures at each stage of their trajectories using intermediate recall and accuracy. Our analysis reveals a consistent pattern: agents often lack awareness of the information they have already collected and whether they have reached a point where information discovery has saturated. As a result, agents continue

issuing search calls and generating reasoning steps while making little or no meaningful progress toward task completion; see Figure 1(a). We refer to this observed pattern as **stagnant reasoning**.

To address this issue, building on our analysis, we hypothesize that DRAs should maintain a broad view of their trajectories, particularly regarding what information has already been covered and whether subsequent steps explore new regions of the search space. This information is necessary for the agent to decide when to continue, redirect/reflect, or stop, preventing it from stagnating. Based on this, we propose a set of unsupervised trajectory-based signals that capture coverage of query criteria and novelty of queries and documents. We integrate these signals into DRAs by introducing **R**etrieval-**A**ware **A**gent **C**ontroller (**RAAC**), an agent-agnostic decision-making controller. At each step of the research process, RAAC leverages these signals to guide the agent to *continue* its reasoning trajectory, *intervene* and adjust its course, or *stop* and generate the final answer. RAAC is agent-agnostic and independent of the design/complexity of the underlying DRA (cf. Figure 6 in Appendix B for an example.)

Our experiments show that applying RAAC to DRAs yields more optimized reasoning trajectories; see Fig. 1(b). Agents with RAAC take fewer steps and execute fewer search calls than those without, lowering computational cost and latency. In particular, RAAC reduces search calls by 14 points on BROWSECOMP-PLUS (Chen et al., 2025) (up to 44), 4 points on NEUCLIR (Lawrie et al., 2025), and 5.5 points on LEGALSEARCH (Korikov et al., 2025). Despite this, agents maintain or improve performance, achieving up to 10% accuracy gains and a 3% average improvement on BROWSECOMP-PLUS. We further analyze the effect of our proposed controller on DRAs through correlation and ablation studies, highlighting the importance of retrieval-aware signals in DRAs.

In summary, our contributions are as follows:

- We conduct, to our knowledge, the first systematic analysis of DRAs based on intermediate behavior, uncovering the ***reasoning stagnation*** problem, where the model lacks awareness of its progress and cannot adapt its search direction or decide when to stop.
- We propose unsupervised trajectory-based signals to mitigate stagnant reasoning and steer DRAs toward more effective and efficient reasoning trajectories.
- We demonstrate the effectiveness of these signals and design a controller that regulates the reasoning process, improving both the efficiency and effectiveness of DRA agents and helping guide future research in DRA optimization.

2 Related Work

Retrieval-augmented generation has evolved from a simple retrieve-then-generate paradigm (Cuconasu et al., 2024; Soudani et al., 2024) to approaches that more tightly integrate retrieval into generation, often interleaving it with intermediate reasoning steps (Trivedi et al., 2023; Jiang et al., 2023; Su et al., 2024; Yao et al., 2023). This shift, further driven by Reinforcement Learning (RL), has led to Deep Research Agents (DRAs), which perform multi-step tool use and extensive information seeking to handle complex queries (Jin et al., 2025). Recent work on improving DRAs has focused on strengthening the search component via retrieval fine-tuning (Chen et al., 2026b), reranking (Sharifymoghaddam and Lin, 2026), and query reformulation (Meng et al., 2026), often leveraging signals from reasoning traces.

A key distinguishing feature of DRAs is their agentic workflow, i.e., how tools are selected, ordered, and used. We group DRAs into four categories based on how this workflow is induced. **The first and second groups**, instruction-tuned (Press et al., 2023; Li et al., 2025c) and RL-trained methods (OpenAI, 2025; GLM Team, 2025; Li et al., 2025b; Chen et al., 2026a; Shao et al., 2025), both follow a ReAct-style loop (Yao et al., 2023), which can be described as:

$$\text{Query} \rightarrow [\text{Think} \rightarrow \text{Search} \rightarrow \text{Observe}]^* \rightarrow \text{Answer},$$

where $*$ denotes that the loop is repeated for as many iterations as the agent deems necessary. **The third group** follows an outline-guided search paradigm (Han et al., 2025). WebWeaver (Li et al., 2025d) introduces a dynamic outline-guided workflow consisting of a planner and a writer. In the planning phase, the model generates an outline along with a memory bank. In the writing phase, the writer retrieves information from the memory bank to produce the final report, as follows:

$$\begin{aligned}&\text{Query} \rightarrow \\ &[\text{Think} \rightarrow \text{Search} \rightarrow \text{Write_outline}]^* \rightarrow \text{Outline} \\ &[\text{Think} \rightarrow \text{Retrieve} \rightarrow \text{Write_section}]^* \rightarrow \text{Report}\end{aligned}$$

The fourth group adopts a report-based generation

style. AgentCPM-Report (Li et al., 2026a) introduces a Writing-as-Reasoning paradigm, in which the report is generated incrementally throughout the search process rather than being preceded by a standalone outline. In addition, the outline (or plan) can be dynamically revised during generation when necessary. The overall workflow is as follows:

Query → Search → Init Plan → [Search → Write]* → [Extend Plan → [Search → Write]*]* → Report

In this paper, we investigate DRAs across all four groups to demonstrate the broad applicability of our findings.

3 Stagnant Reasoning Analysis

We systematically analyze DRAs across iterations to understand their stagnant reasoning behavior. In what follows, we first provide a formal definition of DRAs, then explain the method used for analyzing intermediate search and reasoning behavior of DRAs, and finally report the results of the analysis.

3.1 Deep Research – Task Formulation

Following prior work (Li et al., 2025b,d; Chen et al., 2026b), we formulate DRAs based on the ReAct framework (Yao et al., 2023), which combines reasoning and acting. In this framework, an LLM agent interacts with the environment (e.g., a retriever) over multiple turns. At each turn t, the agent generates both a thought τ_t, representing the agent's internal reasoning process, and a subsequent action a_t, representing an external operation performed to interact with the environment. After executing the action a_t, the agent receives the feedback o_t from the environment. This process forms a trajectory, denoted as $\mathcal{H}_T$, consisting of a sequence of thought–action–observation triplets:

$$\mathcal{H}_T = (\tau_1, a_1, o_1, \cdots, \tau_t, a_t, o_t, \cdots, \tau_T, a_T).$$

Each thought τ_t and action a_t are generated by the agent's policy π, conditioned on a representation of the interaction history $\mathcal{H}_{t-1}$:

$$\tau_t, a_t \sim \pi(\cdot \mid f(\mathcal{H}_{t-1})),$$

where the function f generates a representation of the interaction history, for example by concatenating all text in $\mathcal{H}_{t-1}$ (Yao et al., 2023; Tongyi DeepResearch Team, 2025) or by producing a concise summary of the agent's previous thoughts and observations (Li et al., 2025d, 2026a).

In the most common setting, where the environment consists of a single retriever, each intermediate action corresponds to a search query q_t (i.e., $a_t = q_t$), the final action corresponds to the generated response r_T (i.e., $a_T = r_T$). In this case, the observation o_t consists of the top-k retrieved documents D_t (i.e., $o_t = D_t$), which are obtained by the retrieval model $\mathcal{R}$; i.e., $D_t \leftarrow \mathcal{R}(q_t)$.

3.2 Our Analysis Approach

To better understand the DRAs' decision-making processes, we evaluate their reasoning and search behavior across intermediate turns using a set of evaluation methods described below.

Intermediate Accuracy. DRAs commonly generate the final answer r_T only at the end of the trajectory, after gathering sufficient information. To assess whether a given agent can already answer the user query at iteration t, we spawn a conceptual branch from the main reasoning trajectory by prompting the agent to generate a secondary reasoning step τ'_t, which leads to an intermediate answer r_t (i.e., $a'_t = r_t$) without interfering with the main trajectory. We also allow the agent to output *no candidate* if sufficient information has not yet been gathered. More specifically, two separate trajectories are generated at turn t:

$$\tau'_t, r_t \sim \pi(\cdot \mid f(\mathcal{H}_{t-1})),$$
$$\tau_t, q_t \sim \pi(\cdot \mid f(\mathcal{H}_{t-1})),$$

where r_t is either an answer to the user's question or *no candidate* and τ'_t is the thought leading to the answer r_t. The agent further continues its trajectory with the thought τ_t and query q_t. We compute the intermediate accuracy (Acc_t) by comparing the intermediate response r_t with the ground truth response r^*. See Appendix B for the further details of this metric.

Intermediate Recall. To measure how effectively the retriever discovers relevant information throughout the reasoning trajectory, we define *Intermediate Recall*. At step t, it is defined as the proportion of gold documents that have been retrieved up to that step:

$$\text{R}_t = |D_g \cap D_{1:t}| \,/ |D_g|,$$

where D_g denotes the set of relevant gold documents, and $D_{1:t} = \bigcup_{i=1}^{t} D_i$ represents the union of the documents retrieved up to step t.

Retrieved Document Types. To analyze the reasoning and search behavior of DRAs more precisely, we examine the retrieved documents and categorize them into four types. Formally, each document d_t retrieved in step t belongs to one of the following categories:

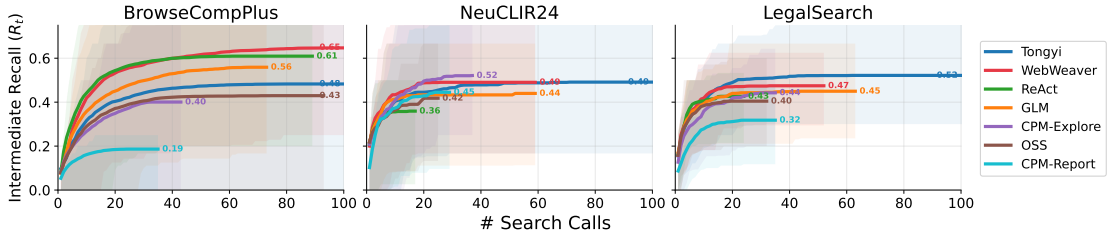

Figure 2: Intermediate Recall (R_t) across seven agents and three datasets, illustrating the stagnation problem in DRAs. Shaded bands indicate the spread (standard deviation) across queries.

RELEVANT NEW: The document is labeled as a gold document and has not been seen before in the trajectory $d_t \in D_g$ and $d_t \notin D_{1:t-1}$.

RELEVANT SEEN: The document is a gold document and has already been observed in the trajectory; i.e., $d_t \in D_g$ and $d_t \in D_{1:t-1}$.

IRRELEVANT NEW: The document is not a gold document and has not been seen before in the trajectory; i.e., $d_t \notin D_g$ and $d_t \notin D_{1:t-1}$.

IRRELEVANT SEEN: The document is not a gold document but has already been observed in the trajectory, i.e., $d_t \notin D_g$ and $d_t \in D_{1:t-1}$.

3.3 Experiments and Results

Deep Research Agents. We evaluate seven DRAs, belonging to four different architectures (cf. Section 2): ReAct (Yao et al., 2023), GLM-4.7 (GLM Team, 2025), GPT-oss-20b-high (OpenAI, 2025), Tongyi-DR (Tongyi DeepResearch Team, 2025), AgentCPM-Explore (Chen et al., 2026a), WebWeaver (Li et al., 2025d), and AgentCPM-Report (Li et al., 2026a). For ReAct and WebWeaver, we use Claude Sonnet 4.5 as the backbone LLM. To ensure fair comparison, all agents use the same retrieval pipeline and receive the top-k retrieved documents per search ($k = 5$), following Chen et al. (2026b); Meng et al. (2026); see Table 3 in Appendix B for further details.

Datasets. We conduct experiments on three datasets: BROWSECOMP-PLUS (Chen et al., 2025), NEUCLIR (Lawrie et al., 2025), and LEGALSEARCH, a closed-source dataset provided by Thomson Reuters (Korikov et al., 2025). All datasets contain human-verified supporting documents, enabling the evaluation of retrieval performance. In addition, BROWSECOMP-PLUS includes short verifiable answers, allowing us to evaluate both intermediate and final generated responses. See Appendix B for further details.

Retriever. Following Chen et al. (2026b); Meng et al. (2026), we employ a single-vector dense retriever using `Qwen3-Embedding-4B` (Qwen Team, 2025) for BROWSECOMP-PLUS and NEUCLIR datasets. For LEGALSEARCH, documents are retrieved through a proprietary multi-stage retrieval pipeline.

Evaluation Metrics. We report intermediate accuracy (Acc_t) and recall (R_t) on BROWSECOMP-PLUS and intermediate recall on other datasets. For *accuracy*, we follow the BROWSECOMP-PLUS evaluation protocol (Chen et al., 2025) and leverage LLM-as-judge comparison of the agent's final answer against the ground truth.

(RQ1) How do DRAs' search and reasoning performances change throughout the reasoning trajectory? Figure 2 presents the intermediate recall for all datasets. While the best performing DRAs varies across these datasets, a consistent pattern emerges across all datasets and agents: after the initial steps, performance quickly saturates, and additional iterations provide little to no meaningful improvement in recall. A similar behavior is observed for intermediate accuracy on BROWSECOMP-PLUS. As shown in Figure 1, most agents (WebWeaver, ReAct, GPT-oss, and CPM-Report) achieve near-final performance after several iterations, with later steps adding little benefit. CPM-Explore also follows the trend of reaching its final accuracy within a few iterations but tends to stop once it finds the correct answer. Only GLM and Tongyi improve gradually, with each iteration contributing consistently to performance.

These results suggest that DRAs often suffer from reasoning stagnation, where agents continue generating steps without making meaningful progress toward task completion. During these reasoning steps, **agents are not fully aware of what they have already gathered**, which constraints have been satisfied, or whether they have already reached the correct final answer. In other words, they lack a clear sense of progress toward the solution. As a result, even after obtaining sufficient information or identifying the correct answer, they may continue invoking tools unnecessarily.

Figure 3: Distribution of document types per reasoning step across the first 20 iterations, showing a reduction in newly retrieved relevant documents.

(RQ2) How does the distribution of document types observed by the agent evolve across iterations? Figure 3 shows the distribution of document types across different agents and datasets for the first 20 iterations of each DRA, which sufficiently captures the observed patterns. We observe a consistent pattern across all settings: over time, agents retrieve fewer new documents (both relevant and irrelevant) and increasingly revisit previously seen ones. Meanwhile, newly retrieved relevant documents steadily decline and eventually disappear after several iterations.

These patterns suggest that **agents struggle to retrieve alternative evidence needed for the final answer**, instead revisiting previously seen relevant documents. This suggests that reasoning stagnation stems from limited exploration of the search space and difficulty escaping local optima defined by the early relevant evidence.

4 Retrieval-Aware Agent Controller

We attribute stagnant reasoning in DRAs to two main causes: (1) limited awareness of information coverage and (2) difficulty shifting criteria or exploring new aspects. We argue that agents need trajectory-level retrieval-aware signals to guide their reasoning process. We propose four unsupervised signals and show that integrating them into DRAs as a controller improves performance.

4.1 Retrieval-Aware Signals

Criteria Coverage (Cov). Each deep research question is commonly associated with a set of criteria $C = [c_1, \ldots, c_M]$, stating what information is needed to fully accomplish the task (Chen et al., 2025; Min et al., 2023).These criteria may be explicitly stated in the query (as for instance provided in BROWSECOMP-PLUS dataset) or otherwise, should be inferred by the agent.

We define Criteria Coverage as the extent to which these criteria have been satisfied over the course of research. The metric reflects whether the agent is effectively exploring the search space, becoming overly focused on a subset of criteria, or has gathered sufficient information to comprehensively answer the question. More specifically, the coverage score at iteration t is defined as:

$$\text{Cov}_t = \frac{1}{|C|} \sum_{c \in C} W(c, t),$$

where $W(c, t)$ is a weighting function representing the coverage of criterion c at iteration t. Following Thakur et al. (2025); Gao et al. (2023), we assign weights of 0.5, 1.0, and 0 to partially covered, fully covered, and uncovered criteria, respectively.

Document Novelty (Nov). This signal measures the fraction of newly retrieved documents in an iteration that have not appeared in any previous iterations. This choice is motivated by the observation in the previous section, where the retrieval of new (relevant) documents declines in the further steps of the trajectory. Novelty at iteration t is defined as the proportion of retrieved documents D_t that do not appear in $D_{1:t-1}$, formulated below.

$$\text{Nov}_t = |D_t \setminus D_{1:t-1}| \,/\, |D_t| \,.$$

Successive Query Diversity (Div). As a complementary signal to *document* novelty, we define a diversity signal for generated *queries*. This signal measures the dissimilarity between the current search query and the query from the previous iteration, and is defined as $\text{Div}_t = 1 - \cos(e_t, e_{t-1})$, where $\cos(\cdot)$ denotes the cosine similarity function, and e_t represents the normalized embedding of query q_t (i.e., $|e_t| = 1$).

Original Question Similarity (Sim). As a guardrail, we define a query similarity signal to ensure that novelty remains aligned with the topic of the user's question and prevents query drift. This guardrail signal is defined as $\text{Sim}_t = \cos(e_t, e_x)$, where e_x denotes the normalized embedding of the user query x. When this similarity becomes too low, it serves as a warning indicator that exploration may be drifting off-topic.

4.2 Agent Controller

We now explain Retrieval-Aware Agent Controller (RAAC) as an approach to incorporate the introduced signals in a given DRA. The core idea of RAAC is that, the controller monitors the agent's interaction history using the proposed signals and adjusts the process of the agent when needed. At each iteration t, the controller computes the retrieval-aware signals (i.e., $\text{Nov}_t, \text{Cov}_t, \text{Div}_t$, and Sim_t), and suggests an action from a predefined set based on the previous turns and their corresponding signals. The action set is defined as $A = \{\textit{continue}, \textit{intervene}, \textit{stop}\}$, and explained in what follows. This overall architecture of RAAC is depicted in Figure 6 in Appendix B.

Continue. This action simply allows the agent to proceed with no intervention.

Intervene. As discussed in previous studies (Soudani et al., 2025; Li et al., 2025a), when the trajectory becomes stale but still recoverable, an external nudge (e.g., critical reasoning) can redirect the search toward a more informative direction. Upon invoking *Intervene*, the controller calls a critical re-thinker, which (1) critically assesses why the current search trajectory is stagnating, and (2) generates a substantially different and potentially creative search query to help escape the loop. More specifically, the controller takes $\mathcal{H}_t$ as input and generates a revised reasoning trace τ^{cr}_{t+1} together with a new search query q^{cr}_{t+1}. The corresponding document set D^{cr}_{t+1} is then retrieved, and the resulting triplet is appended to the interaction history:

$$\mathcal{H}_{t+1} = \mathcal{H}_t \oplus \left(\tau^{cr}_{t+1}, q^{cr}_{t+1}, D^{cr}_{t+1}\right).$$

Stop. This action is selected when the trajectory is exhausted and further iterations are unlikely to yield new information (Park et al., 2025; Hu et al., 2025). Given $\mathcal{H}_t$, the agent generates a final reasoning step τ'_T, which leads to the final answer r_T, defined as: $\tau'_T, r_T \sim \pi(\cdot \mid f(\mathcal{H}_t))$.

Agent	BrowseCompP.			NeuCLIR		LegalSearch	
	R	Acc	#S↓	R	#S↓	R	#S ↓
CPM-Report	18.6	11.7	14.4	44.5	20.6	31.8	19.4
+ RAAC	**30.5**†	**12.0**	18.2	**48.7**	22.2	**40.8**†	21.6
GPT-oss-20b	43.0	23.0	28.3	41.8	12.2	40.5	10.2
+ RAAC	**50.7**†	**26.4**†	27.0	**43.7**	14.0	**43.8**	10.8
CPM-Explore	40.1	**30.5**	29.6	**52.1**	25.5	44.2	23.8
+ RAAC	**48.0**†	29.7	21.7	42.5†	12.5	**45.2**	12.2
GLM-4.7	55.9	32.0	50.3	43.9	12.1	45.0	19.8
+ RAAC	**59.2**†	**34.9**†	36.8	**46.8**	17.1	**50.0**	18.6
ReAct	60.9	35.7	35.3	35.9	5.6	42.6	8.7
+ RAAC	**62.6**	**39.1**	25.4	**39.1**	5.7	**47.5**†	8.4
WebWeaver	64.9	47.8	75.0	49.0	19.2	47.5	19.9
+ RAAC	**67.2**†	**51.3**†	50.7	**50.0**	20.3	**51.6**	22.1
Tongyi-DR	48.4	37.9	77.8	49.1	62.1	52.2	53.8
+ RAAC	**59.0**†	**47.9**†	33.6	**55.6**†	34.4	**56.5**	22.8

Table 1: Performance of DRAs without and with the RAAC. **R**: recall, **Acc**: accuracy, and **#S**: number of search calls. The numbers in **bold** and with underline show the best results for an agent and overall, respectively. Significant differences are marked with †. Overall trend: enhancing DRAs with RAAC improves the performance of the agents and simultaneously decreases the number of search calls.

4.3 Experiments and Results

We follow the same experimentation setup as explained in Section 3.3. To calculate Cov, the required criteria are provided in BROWSECOMP-PLUS. For NEUCLIR (Lawrie et al., 2025) and LEGALSEARCH, we implement an LLM-based component to create these criteria and further update them (details provided in Appendix B). For the signals which require cosine similarity, we use `Qwen3-Embedding-4B` as the encoder. Besides the overall accuracy, we report (overall) recall, which measures the proportion of relevant documents returned across *all* search calls for a query (Meng et al., 2026). Statistical significance is reported using t-test for recall and McNemar test for accuracy ($p < 0.05$). Appendix B provides the details of the experimentation setup and applied prompts.

(RQ3) What is the effect of RAAC in mitigating reasoning stagnation? The evaluation results of the agents with and without RAAC are reported in Table 1. Since BROWSECOMP-PLUS contains gold criteria, all four signals of Cov_t, Nov_t, Div_t, Sim_t. On NEUCLIR and LEGALSEARCH, RAAC uses the same signals except Cov_t.

Focusing first on the performance metrics, namely recall on all datasets and accuracy on

Ag.	Sig.	BrowseCompP.			NeuCLIR		LegalSearch	
		R	Acc	#S↓	R	#S↓	R	#S ↓
OSS	*N-C-D*	**50.7**	**26.4**	27.0	39.3	11.6	40.2	10.0
	N-D	48.5†	24.5†	25.5	**43.7**	14.0	**43.8**	10.8
	C-D	41.4†	17.2†	12.3	39.3	10.4	40.7	8.9
GLM	*C-N-D*	**59.2**	**34.9**	36.8	44.1	12.0	47.2	13.9
	N-D	56.3†	33.9	35.3	**46.8**	17.1	**50.0**	18.6
	C-D	45.1†	26.9†	14.7	41.1†	10.0	44.4†	11.2
WebWeaver	*C-N-D*	**67.2**	51.3	50.7	47.9	19.8	50.1	19.1
	N-D	65.7	**51.8**	42.1	**50.0**	20.3	**51.6**	22.1
	C-D	56.7†	43.2†	25.6	46.5	16.9	46.2	16.4
Tongyi	*C-N-D*	**59.0**	47.9	33.6	47.9†	21.1	49.7†	14.4
	N-D	56.9	**49.1**	31.3	**55.6**	34.4	**56.5**	22.8
	C-D	48.1†	45.6†	18.1	47.8†	16.9	43.6†	12.1

Table 2: Evaluation of the agents with RAAC under different sets of signals. C, N, and D denote Criteria Coverage (Cov), Document Novelty (Nov), and Query Diversity (Div), respectively. Significant differences are compared against the gray row in each block and marked with †.

BROWSECOMP-PLUS. The results show that, across all agents after applying RAAC, the performance metrics either improve or remain on par. From an efficiency point of view, we observe a reduction in the number of search calls / iterations on BROWSECOMP-PLUS for all agents except CPM-Report. Interestingly, the increase in the number of search calls for CPM-Report leads to a significant improvement in both performance metrics, as the original agent (in contrast to the others) inherently tends to terminate too early, corrected by RAAC.

A similar pattern is observed in NEUCLIR and LEGALSEARCH. On these datasets, applying RAAC mostly reduces the number of search calls. In cases where the number of iterations increases, we consistently observe an increase in performance, showing the adaptive behavior of RAAC to iterate more when more evidence is needed.

On all datasets, the best overall performance is achieved by applying RAAC to the previous SOTA agents. In particular, on BROWSECOMP-PLUS WebWeaver+RAAC achieves the best results by significantly improving WebWeaver on both recall and accuracy, and simultaneously decreasing the number of its search calls by 33%. On NEUCLIR, Tongyi-DR+RAAC achieves the best overall recall and significantly improves its base agent, while decreasing the average search calls by 45%. The same pattern is observed on LEGALSEARCH with an improvement in recall followed by 58% decrease in the number of iterations.

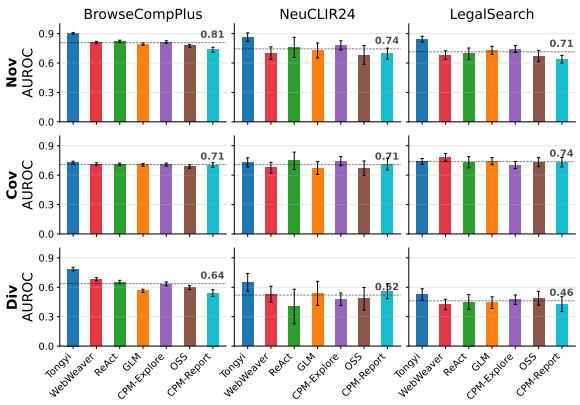

Figure 4: The correlation between unsupervised signals and New-item Recall, measured using AUROC.

To summarize, applying RAAC to the DRAs under the study in most cases reduces unnecessary iterations through targeted interventions, and in some cases increases the iterations to accumulate more evidence. The results demonstrate that the importance of signal-driven controller for DRAs to improve efficiency and effectiveness and to support with robust adaptivity across different trajectories.

(RQ4) What is the effect of each signal on controlling DRAs behavior? Table 2 presents the results of an ablation study of the controller under different signal combinations. In this table, we denote Cov as C, Nov as N, and Div as D. We keep Sim in all setups, as it is defined as the guardrail signal. We report results for four agents due to context limitations. Full results are in Table 5.

On BROWSECOMP-PLUS, incorporating Cov improves performance, whereas on NEUCLIR and LEGALSEARCH, better results are achieved without it. This is due to the presence of human-generated criteria in BROWSECOMP-PLUS queries and suggests that LLM-generated criteria are useful only when human-generated criteria are not available in the query. Across all datasets, removing Nov as the primary signal reduces recall, indicating that Nov is the most effective choice for the primary signal.

To further analyze the informativeness of each signal (independent of its integration in RAAC), we compute the correlation between each signal and recall. We define ***New-item Recall*** as the fraction of new relevant documents retrieved at iteration t that have not been retrieved in previous iterations. Formally,

$$\text{NiR}_t = \frac{|(D_t \cap D_g) \setminus D_{1:t-1}|}{|D_g|}.$$

We argue that the correlation between unsuper-

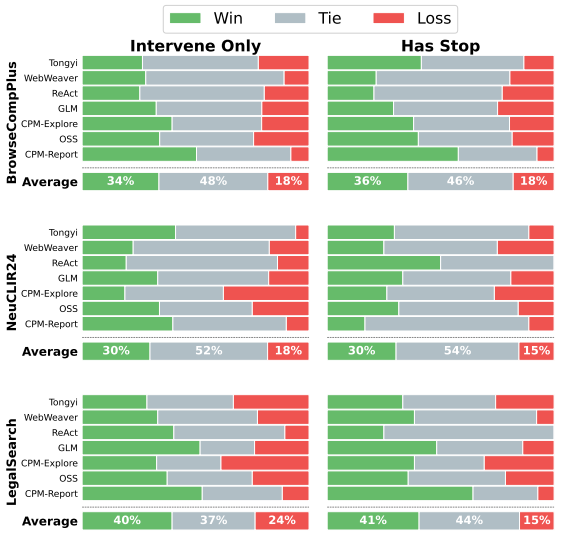

Figure 5: Evaluation of the impact of the controller's actions measured by recall.

vised signals and *New-item Recall* serves as a reliable proxy for measuring the informativeness of these signals. Therefore, we evaluate how well each signal discriminates between productive iterations ($\text{NiR}_t > 0$) and non-productive iterations ($\text{NiR}_t = 0$). Because PiR is *zero-inflated*, with most iterations failing to retrieve any new relevant documents, we report AUROC, as it quantifies how well a signal distinguishes between productive and non-productive iterations, making it particularly appropriate for sparse binary-like outcomes.

Figure 4 presents the correlation between PiR and each signal (apart from the guardrail Sim_t signal) across all setups. On average, Nov_t exhibits the highest correlation, followed by Cov_t, while Div_t shows the lowest correlation. These results are largely consistent with the roles in the controller. Therefore, the correlation with PiR_t serves as a valid proxy for measuring the effectiveness of the unsupervised signals, highlighting the importance of Nov_t and Cov_t as key signals to control stagnant reasoning in DRAs.

(RQ5) What is the effect of RAAC actions on steering the trajectory toward an optimal path? We analyze the effect of the two active controller actions, intervene and stop. We partition the queries into two groups based on the controller's behavior during the trajectory: *Has Stop*, which contains queries for which the controller issued the Stop action, and *Intervene Only*, which contains queries where the controller *only* issued Intervene action and never stopped the agent early. Within each group, we compute the final number of retrieved gold documents with and without RAAC and categorize queries to *win* (more gold documents with controller), *loss* (fewer gold documents), or a *tie* (equal count) group.

Figure 5 presents the win/tie/loss distributions across all agents and datasets. In the *Has Stop* group, wins consistently outnumber losses, with average win rates of 35% vs. 18% on BROWSECOMP-PLUS, 30% vs. 15% on NEUCLIR, and 41% vs. 15% on LEGALSEARCH. The *Intervene Only* group shows a similar positive trend (33% vs. 18%, 30% vs. 18%, and 40% vs. 24%, respectively). These results suggest that both controller actions are effective: intervention helps redirect trajectories toward higher-recall paths, while stopping does not prematurely terminate productive iterations.

We further analyze how samples are distributed across groups. On BROWSECOMP-PLUS, 61.5% of query–agent pairs fall into the *Has Stop* group and 34.6% into the *Intervene Only* group, indicating that the controller actively intervenes in the majority of trajectories. In contrast, on NEUCLIR and LEGALSEARCH, the pattern reverses: *Intervene Only* dominates (55.4% and 54.8%, respectively). This reflects the shorter trajectories in these datasets, where the controller more often redirects behavior than terminates it.

5 Discussion and Conclusions

In this paper, we analyze the reasoning behavior of DRAs through intermediate recall and accuracy, revealing that these agents suffer from stagnant reasoning, where many iterations yield little to no meaningful performance improvement. We then argue that agents should be informed about the progress of their reasoning trajectory, which can be achieved through trajectory-based signals. To this end, we propose four unsupervised signals: document novelty, criteria coverage, search query diversity, and original query similarity. Based on these signals, we design a controller that selects an action from a predefined set (continue, intervene, or stop) in order to redirect the reasoning trajectory toward a more optimal path. Our experimental results across seven agents and three datasets show that the controller consistently improves effectiveness while reducing latency, validating the usefulness of the proposed signals. A promising direction for future work is to train agents that are inherently aware of unsupervised trajectory signals during task completion.

Limitations

Prompt-based Controller. Our controller (Section 4.2) relies on a hand-crafted prompt to map trajectory signals into one of three discrete actions: continue, intervene, or stop. While this design is effective and requires no training data, it applies fixed weights to these signals uniformly across all queries, rather than adapting them to the characteristics of individual queries. A learned controller could instead adaptively weight the signals, capture dependencies across successive observations, and potentially generalize better to diverse query types. We leave the exploration of learned controllers, including supervised and reinforcement learning-based approaches, to future work.

Comprehensiveness of the Signals. The controller leverages unsupervised trajectory signals to estimate task completion progress. While we introduce four signals and analyze the contribution of each, we are interested in identifying additional signals to make the decision-making process more informed. For example, signals derived from document content, retrieval confidence scores, or inter-document redundancy may provide complementary information. Future work could explore a broader range of trajectory signals and investigate their interactions.

Comprehensiveness of the Actions. Our analysis identifies two recurring failure modes in DRAs: failing to stop at the optimal point and becoming trapped in a narrow search space without exploring alternative perspectives. To address these issues, we introduce two controller actions: stop and intervene (through critical thinking), alongside a continue action that preserves the agent's default behavior. Although these actions target the most prominent shortcomings observed in our analysis, DRAs are a rapidly evolving paradigm and may exhibit additional failure modes as they are applied to a broader range of tasks. Future work could explore these limitations further and introduce additional actions to address them.

References

Haotian Chen, Xin Cong, Shengda Fan, Yuyang Fu, Ziqin Gong, Yaxi Lu, Yishan Li, Boye Niu, Chengjun Pan, Zijun Song, Huadong Wang, Yesai Wu, Yueying Wu, Zihao Xie, Yukun Yan, Zhong Zhang, Yankai Lin, Zhiyuan Liu, and Maosong Sun. 2026a. Agentcpm-explore: Realizing long-horizon deep exploration for edge-scale agents. *CoRR*, abs/2602.06485.

Zijian Chen, Xueguang Ma, Shengyao Zhuang, Jimmy Lin, Akari Asai, and Victor Zhong. 2026b. AgentIR: Reasoning-aware retrieval for deep research agents. *arXiv preprint arXiv:2603.04384*.

Zijian Chen, Xueguang Ma, Shengyao Zhuang, Ping Nie, Kai Zou, Andrew Liu, Joshua Green, Kshama Patel, Ruoxi Meng, Mingyi Su, Sahel Sharifymoghaddam, Yanxi Li, Haoran Hong, Xinyu Shi, Xuye Liu, Nandan Thakur, Crystina Zhang, Luyu Gao, Wenhu Chen, and Jimmy Lin. 2025. BrowseComp-Plus: A more fair and transparent evaluation benchmark of deep-research agent. *arXiv preprint arXiv:2508.06600*.

Florin Cuconasu, Giovanni Trappolini, Federico Siciliano, Simone Filice, Cesare Campagnano, Yoelle Maarek, Nicola Tonellotto, and Fabrizio Silvestri. 2024. The power of noise: Redefining retrieval for RAG systems. In *Proceedings of the 47th International ACM SIGIR Conference on Research and Development in Information Retrieval, SIGIR*, pages 719–729.

Tianyu Gao, Howard Yen, Jiatong Yu, and Danqi Chen. 2023. Enabling large language models to generate text with citations. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing, EMNLP*, pages 6465–6488. Association for Computational Linguistics.

GLM Team. 2025. GLM-4.5: Agentic, reasoning, and coding (ARC) foundation models. *arXiv preprint arXiv:2508.06471*.

Rujun Han, Yanfei Chen, Zoey CuiZhu, Lesly Miculicich, Guan Sun, Yuanjun Bi, Weiming Wen, Hui Wan, Chunfeng Wen, Solène Maître, George Lee, Vishy Tirumalashetty, Emily Xue, Zizhao Zhang, Salem Haykal, Burak Gokturk, Tomas Pfister, and Chen-Yu Lee. 2025. Deep researcher with test-time diffusion. *CoRR*, abs/2507.16075.

Yunhai Hu, Yilun Zhao, Chen Zhao, and Arman Cohan. 2025. MCTS-RAG: enhancing retrieval-augmented generation with monte carlo tree search. In *Findings of the Association for Computational Linguistics: EMNLP*, pages 12581–12597. Association for Computational Linguistics.

Soyeong Jeong, Jinheon Baek, Sukmin Cho, Sung Ju Hwang, and Jong Park. 2024. Adaptive-RAG: Learning to adapt retrieval-augmented large language models through question complexity. In *Proceedings of the 2024 Conference of the North American Chapter of the Association for Computational Linguistics, NAACL*.

Zhengbao Jiang, Frank Xu, Luyu Gao, Zhiqing Sun, Qian Liu, Jane Dwivedi-Yu, Yiming Yang, Jamie Callan, and Graham Neubig. 2023. Active retrieval augmented generation. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 7969–7992.

Bowen Jin, Hansi Zeng, Zhenrui Yue, Dong Wang, Hamed Zamani, and Jiawei Han. 2025. Search-r1: Training llms to reason and leverage search engines with reinforcement learning. *CoRR*.

Anton Korikov, Pan Du, Scott Sanner, and Navid Rekabsaz. 2025. Batched self-consistency improves LLM relevance assessment and ranking. *arXiv preprint arXiv:2505.12570*.

Dawn Lawrie, Sean MacAvaney, James Mayfield, Paul McNamee, Douglas W. Oard, Luca Soldaini, and Eugene Yang. 2025. Overview of the TREC 2024 NeuCLIR track. *arXiv preprint arXiv:2509.14355*.

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*, Findings of ACL, pages 11258–11276. Association for Computational Linguistics.

Baixuan Li, Bo Zhang, Dingchu Zhang, Fei Huang, Guangyu Li, Guoxin Chen, Huifeng Yin, Jialong Wu, Jingren Zhou, Kuan Li, Liangcai Su, Litu Ou, Liwen Zhang, Pengjun Xie, Rui Ye, Wenbiao Yin, Xinmiao Yu, Xinyu Wang, Xixi Wu, and 36 others. 2025b. Tongyi deepresearch technical report. *CoRR*, abs/2510.24701.

Xiaoxi Li, Guanting Dong, Jiajie Jin, Yuyao Zhang, Yujia Zhou, Yutao Zhu, Peitian Zhang, and Zhicheng Dou. 2025c. Search-o1: Agentic search-enhanced large reasoning models. In *Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing, EMNLP*, pages 5420–5438. Association for Computational Linguistics.

Yishan Li, Wentong Chen, Yukun Yan, Mingwei Li, Sen Mei, Xiaorong Wang, Kunpeng Liu, Xin Cong, Shuo Wang, Zhong Zhang, Yaxi Lu, Zhenghao Liu, Yankai Lin, Zhiyuan Liu, and Maosong Sun. 2026a. AgentCPM-Report: Interleaving drafting and deepening for open-ended deep research. *arXiv preprint arXiv:2602.06540*.

Zhuofeng Li, Haoxiang Zhang, Cong Wei, Pan Lu, Ping Nie, Yi Lu, Yuyang Bai, Shangbin Feng, Hangxiao Zhu, Ming Zhong, and 1 others. 2026b. Beyond semantic similarity: Rethinking retrieval for agentic search via direct corpus interaction. *arXiv preprint arXiv:2605.05242*.

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

Alex Mallen, Akari Asai, Victor Zhong, Rajarshi Das, Daniel Khashabi, and Hannaneh Hajishirzi. 2023. When not to trust language models: Investigating effectiveness of parametric and non-parametric memories. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 9802–9822.

Chuan Meng, Litu Ou, Sean MacAvaney, and Jeff Dalton. 2026. Revisiting text ranking in deep research. *arXiv preprint arXiv:2602.21456*.

Sewon Min, Kalpesh Krishna, Xinxi Lyu, Mike Lewis, Wen-tau Yih, Pang Wei Koh, Mohit Iyyer, Luke Zettlemoyer, and Hannaneh Hajishirzi. 2023. Factscore: Fine-grained atomic evaluation of factual precision in long form text generation. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing, EMNLP*, pages 12076–12100. Association for Computational Linguistics.

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

Jaewan Park, Solbee Cho, and Jay-Yoon Lee. 2025. Stop-rag: Value-based retrieval control for iterative rag. *arXiv preprint arXiv:2510.14337*.

Ofir Press, Muru Zhang, Sewon Min, Ludwig Schmidt, Noah Smith, and Mike Lewis. 2023. Measuring and narrowing the compositionality gap in language models. In *Findings of the Association for Computational Linguistics: EMNLP*.

Qwen Team. 2025. Qwen3 Embedding: Advancing text embedding and reranking through foundation models. *arXiv preprint arXiv:2506.05176*.

Alireza Salemi, Mukta Maddipatla, and Hamed Zamani. 2025. Ciir@ liverag 2025: Optimizing multi-agent retrieval augmented generation through self-training. *arXiv preprint arXiv:2506.10844*.

Rulin Shao, Akari Asai, Shannon Zejiang Shen, Hamish Ivison, Varsha Kishore, Jingming Zhuo, Xinran Zhao, Molly Park, Samuel G. Finlayson, David A. Sontag, Tyler Murray, Sewon Min, Pradeep Dasigi, Luca Soldaini, Faeze Brahman, Wen-tau Yih, Tongshuang Wu, Luke Zettlemoyer, Yoon Kim, and 2 others. 2025. DR tulu: Reinforcement learning with evolving rubrics for deep research. *CoRR*, abs/2511.19399.

Sahel Sharifymoghaddam and Jimmy Lin. 2026. Rerank before you reason: Analyzing reranking tradeoffs through effective token cost in deep search agents. *arXiv preprint arXiv:2601.14224*.

Heydar Soudani, Evangelos Kanoulas, and Faegheh Hasibi. 2024. Fine tuning vs. retrieval augmented generation for less popular knowledge. In *Proceedings of the 2024 Annual International ACM SIGIR Conference on Research and Development in Information Retrieval in the Asia Pacific Region, SIGIR-AP 2024*, pages 12–22.

Heydar Soudani, Hamed Zamani, and Faegheh Hasibi. 2025. Uncertainty quantification for retrieval-augmented reasoning. *CoRR*, abs/2510.11483.

Weihang Su, Yichen Tang, Qingyao Ai, Zhijing Wu, and Yiqun Liu. 2024. DRAGIN: Dynamic retrieval augmented generation based on the real-time information needs of large language models. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 12991–13013.

Nandan Thakur, Ronak Pradeep, Shivani Upadhyay, Daniel Campos, Nick Craswell, and Jimmy Lin. 2025. Support evaluation for the TREC 2024 RAG track: Comparing human versus LLM judges. *CoRR*, abs/2504.15205.

Tongyi DeepResearch Team. 2025. Tongyi DeepResearch technical report. *arXiv preprint arXiv:2510.24701*.

Harsh Trivedi, Niranjan Balasubramanian, Tushar Khot, and Ashish Sabharwal. 2023. Interleaving retrieval with chain-of-thought reasoning for knowledge-intensive multi-step questions. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 10014–10037.

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

Appendix

A Related Work

Retrieval Control: Signal-driven controllers have recently been proposed for dynamic RAG systems. FLARE (Jiang et al., 2023) and DRAGIN (Su et al., 2024) trigger retrieval based on token-level confidence, Adaptive-RAG (Jeong et al., 2024) uses a complexity classifier for routing, and Stop-RAG (Park et al., 2025) learns when to stop retrieval via a finite-horizon MDP. However, these methods typically rely on a single signal and support only binary decisions. In contrast, our DRA controller integrates multiple trajectory-level signals and enables a richer action space: continue, intervene, or stop.

B Experimental Setup

Intermediate Accuracy. Section 3.2 introduces Intermediate Accuracy. In this section, we provide additional details on its implementation. To this end, we adopt a forced answer generation strategy inspired by the Tongyi agent (Li et al., 2025b), which appends a special instruction to the conversation once the context limit is reached, prompting the agent to produce a final answer. We apply this strategy in our evaluation setup. First, we design a candidate generation instruction explicitly designed to prompt the agent to produce an answer (Figure 7). Since each agent outputs responses in a specific format, we also design an agent-specific format instruction to ensure valid outputs. Figure 8 illustrates the output format for the ReAct agent. Crucially, the instruction allows the model to respond with *no candidate* when it determines that the accumulated evidence is insufficient to produce an answer, thereby avoiding hallucinated guesses in early iterations when only limited information has been retrieved.

It is important to note that we observe different behaviors across agents when they are forced to generate an answer during intermediate reasoning. As a result, for some agents such as WebWeaver, we introduce stronger instructions to enforce compliance (Figure 9).

Evaluation Metrics. For the evaluation, we follow prior work on DRAs (Chen et al., 2026b; Meng et al., 2026; Li et al., 2026b) and report the number of search calls, recall, and accuracy (see Tables 1 and 2). In DRAs, the number of documents seen by the agent, N, is adaptive and varies across queries. We argue that reporting N, Precision@N, and F1@N is meaningful; however, results across different setups are not directly comparable due to variations in N. These metrics are defined in DRAs as follows:

- ***N***: the number of documents the agent has actually seen for a given query.
- ***Recall@N***: measures the fraction of all known relevant documents that the agent successfully seen. A higher value means the agent is more complete in finding relevant documents, it misses fewer.
- ***Precision@N***: measures the fraction of the agent's retrieved seen documents that are actually relevant. A higher value means the agent is more selective, it retrieves less noise and wastes fewer retrieval steps on irrelevant documents.
- ***F1@N***: is the harmonic mean of Recall and Precision, balancing both into a single score. A higher value means the agent achieves a good trade-off between completeness and selectivity simultaneously.

Table 5 reports Precision@N, F1@N, and N alongside the main metrics.

Datasets. We conduct experiments on three datasets. BROWSECOMP-PLUS (Chen et al., 2025), a widely used benchmark for deep research with short, verifiable answers, contains 830 fact-seeking, reasoning-intensive queries paired with concise objective answers. The corpus contains 100K documents.

We further evaluate on two Information Retrieval (IR) datasets: NEUCLIR (Lawrie et al., 2025) and LEGALSEARCH (Korikov et al., 2025). These datasets consist of complex questions that require retrieving and synthesizing information from multiple relevant documents, rather than relying on a single document, to derive the final answer. NEUCLIR (Lawrie et al., 2025) is a TREC benchmark for cross-lingual information retrieval. It contains 74 complex topic descriptions over a corpus of approximately 10M documents originally written in Chinese, Russian, and Persian. In this study, we use the English translations of the documents provided by the dataset creators. LEGALSEARCH is a closed-source dataset from Thomson Reuters comprising

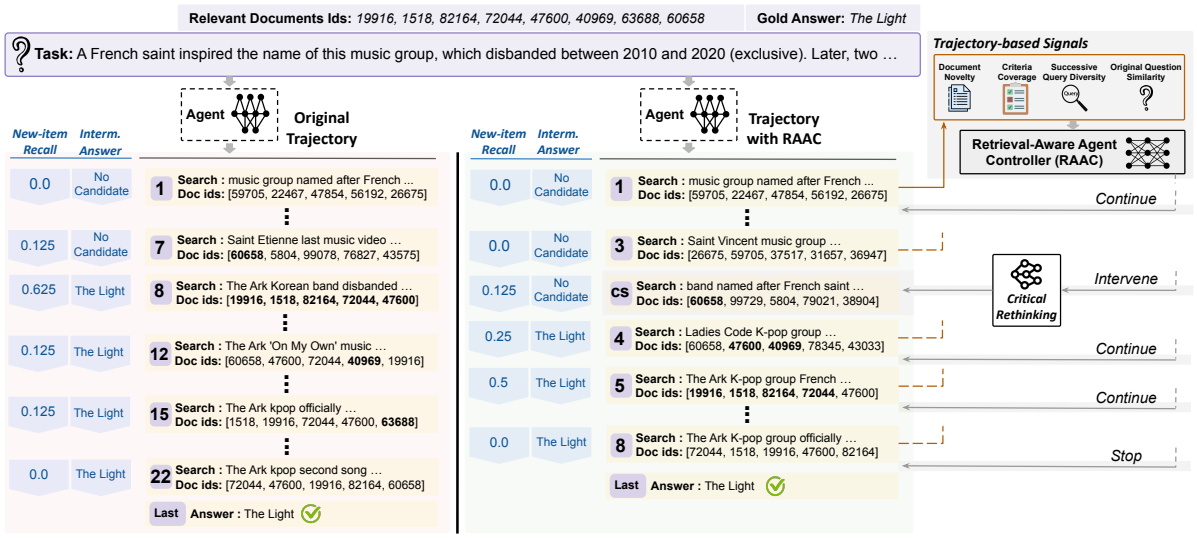

Figure 6: Comparison of the trajectories of a given query between an arbitrary Deep Research Agent (DRA) with and without Retrieval-Aware Agent Controller (RAAC). In the original trajectory, the agent answers the query in 23 iterations, although the answer becomes available at iteration 8. With RAAC, the agent finds a relevant document earlier, making the answer available at iteration 4, and the controller stops the process at iteration 8.

Agent	Backbone / Model	Size	Temp.	Top-p	Max Model Len	Max Num Seqs
AgentCPM-Report	AgentCPM-Report	7B	0.7	1.0	65,536	64
GPT-oss-20B	gpt-oss-20b	20B	0.0	1.0	131,072	16
AgentCPM-Explore	AgentCPM-Explore	4B	1.0	1.0	32,768	64
GLM-4.7	GLM-4.7-Flash	30B	0.0	1.0	65,536	16
ReAct	Claude Sonnet 4.5	–	0.0	1.0	–	–
WebWeaver	Claude Sonnet 4.5	–	0.0	1.0	–	–
Tongyi-DR	Tongyi-DR-30B-A3B	30B	0.6	0.95	131,072	16

Table 3: Inference hyperparameters for each deep research agent. **Size**: backbone parameter count; **Temp.**: sampling temperature; **Max Model Len** and **Max Num Seqs**: vLLM (v0.19.1) serving configuration (dashes indicate API-served models). All agents use top-k=5 retrieval and run for at most 100 iterations.

Answer Candidate Generation

You have now reached the maximum context length you can handle. You should stop making tool calls and, based on all the information above, think again and provide what you consider the most likely answer.
If the evidence is insufficient to answer, respond with 'no candidate'.

Figure 7: Intermediate Answer Prompt.

Answer Candidate Generation - Format

Provide your answer in the following format:
<think>your final thinking</think>\n"
<action>finish(answer="your answer")</action>

Figure 8: Intermediate Answer Prompt ReAct, Formant Instruction.

100 legal queries and paragraph-level passages segmented from approximately 100K legal documents. It has also been used in Korikov et al. (2025). The dataset statistics are summarized in Table 4.

Signals and Controller. For Section 4, we design a prompt-based criteria coverage updater, controller, and critical re-thinker for intervention actions. In this section, we provide more details about the implementation.

CRITERIA COVERAGE: For each query, the signal component first generates a list of criteria using the initialization prompt (Figure 10). For the BrowseCompPlus dataset, the initialization phase is more straightforward because the criteria are explicitly provided in the user query. Therefore, the component does not need to generate criteria from its

Answer Candidate Generation - Format

Based on the evidence gathered so far, provide a concise answer. IMPORTANT: The <answer> tag must contain ONLY the short factual answer,no explanations, no reasoning, no bullet points. Put all reasoning inside <think>.

<think> your reasoning over the collected evidence </think>
<answer> your short answer </answer>

Figure 9: Intermediate Answer Prompt WebWeaver, Formant Instruction.

Dataset	#Queries	Avg. Gold	Corpus Size
BrowseComp-Plus	830	6.1	100K
NeuCLIR 2024	74	6.4	10M
LegalSearch	100	10.4	–

Table 4: Dataset statistics. **Avg. Gold** denotes the average number of gold documents per query.

internal knowledge and only extracts and lists the existing criteria. During the trajectory, whenever the component is invoked, it takes the documents retrieved in the current iteration and updates the status of each criterion. In addition, the component can modify the criteria list by adding or removing criteria. The update prompt is shown in Figure 11. For the BrowseCompPlus dataset, the update process is again more straightforward because the number of criteria is fixed, and only the status of each criterion needs to be updated.

Criteria Coverage - Initialization

You are an aspect-coverage assessor for an iterative deep-research agent.
You will receive a query and must decompose it into its key information-need aspects.

== Rules ==
1. Identify {min_aspects} to {max_aspects} distinct aspects that a comprehensive answer must address.
2. Each aspect should be independently answerable and roughly equal in scope — not too abstract, not too specific.
3. Minimize overlap — each aspect should cover territory not addressed by any other.

== Output Format ==
Respond with ONLY a raw JSON object — no markdown, no code blocks.

Keys:
- "reasoning": State which facets of the query you identified and why you chose this decomposition.
- "aspects": A list of objects, each with:
* - "name": A short, descriptive name for the aspect.*
* - "status": "not_covered"*
* - "evidence": ""*
- "critical_gaps": A list of all aspect names.

Example for the query "How can the Internet of Things (IoT) help cities become smarter and more efficient?":
{{
"reasoning": "The query asks how IoT enables smarter cities. I identified six distinct facets: the foundational connectivity layer, concrete application domains, safety infrastructure, revenue opportunities, governance frameworks, and the human capital needed to sustain it all.",
"aspects": [
{{"name": "IoT device connectivity", "status": "not_covered", "evidence": ""}},
{{"name": "smart city applications", "status": "not_covered", "evidence": ""}},
{{"name": "fire prevention and safety systems", "status": "not_covered", "evidence": ""}},
{{"name": "municipal revenue generation", "status": "not_covered", "evidence": ""}},
{{"name": "infrastructure and governance systems", "status": "not_covered", "evidence": ""}},
{{"name": "government funding and workforce training", "status": "not_covered", "evidence": ""}}
],
"critical_gaps": ["IoT device connectivity", "smart city applications", "fire prevention and safety systems", "municipal revenue generation", "infrastructure and governance systems", "government funding and workforce training"]
}}

Figure 10: Criteria Coverage Initialization Prompt.

Criteria Coverage - Update

You are an aspect-coverage assessor for an iterative deep-research agent.
After each search iteration you receive the query, current aspects, retrieved documents, and search queries. Your job is to update aspect statuses and, when warranted, restructure the aspect list.

== Aspect Statuses ==
- "not_covered": No retrieved document contains information relevant to this aspect.
- "partial": At least one document mentions this aspect, but the information is indirect, incomplete, or insufficient to answer the aspect's need.
- "covered": At least one document directly and sufficiently addresses the aspect's information need.

== Actions ==
- "tick": Update an aspect's status or evidence. Use when new evidence supports an upgrade, adds detail, or contradicts earlier conclusions.
- "add": Add a new aspect discovered from the evidence, when it represents a genuinely distinct information need not captured by any existing aspect.
- "remove": Remove an aspect that is clearly redundant with another or irrelevant to the query.

== Rules ==
1. For each new document, check every aspect: does this document contain information relevant to the aspect?
2. Upgrade aggressively: if a document directly answers an aspect's need, set it to "covered" — do not require multiple sources.
3. Upgrade to "partial" when a document is tangentially relevant or provides only a clue toward the aspect.
4. Downgrade only when new evidence contradicts earlier conclusions.
5. Only return actions for aspects that changed. The system applies your actions as a delta.
6. Maximum {max_aspects} aspects. If you want to add beyond the cap, merge with an existing aspect instead.

== Output Format ==
Respond with ONLY a raw JSON object — no markdown, no code blocks.

Keys:
- "reasoning": For each aspect that changed, state what evidence caused the change and why.
- "actions": A list of action objects. Each object has:
* - "action": One of "tick", "add", "remove".*
* - "name": The aspect name (must match an existing name for "tick" and "remove").*
* - "status": The new status (required for "tick" and "add", omit for "remove").*
* - "evidence": Brief note citing the source number(s) and what they say (required for "tick" and "add", omit for "remove").*
- "critical_gaps": Aspect names that are still "not_covered", ordered by importance.
- "minor_gaps": Aspect names that are still "partial", ordered by importance.

Example for the query "How can the Internet of Things (IoT) help cities become smarter and more efficient?":
{{
"reasoning": "New documents provided detailed case studies on smart city applications from three independent cities, upgrading from not_covered to partial. Discovered data privacy as a distinct information need not captured by existing aspects. Removing 'municipal revenue generation' as it overlaps with 'infrastructure and governance systems'.",
"actions": [
{{"action": "tick", "name": "smart city applications", "status": "partial", "evidence": "Case studies from Barcelona, Singapore, and Copenhagen describe traffic, waste, and lighting systems (Sources 1, 4, 6)"}},
{{"action": "add", "name": "data privacy and security", "status": "partial", "evidence": "One source highlights citizen data collection risks in sensor networks (Source 4)"}},
],
"critical_gaps": ["IoT device connectivity", "government funding and workforce training"],
"minor_gaps": ["smart city applications", "data privacy and security"]
}}

Figure 11: Criteria Coverage Update Prompt.

CRITICAL RE-THINKER: We instantiate the intervention action as a critical rethinking step implemented by an LLM-based component. Figure 12 shows the prompt, which is designed based on (Soudani et al., 2025; Salemi et al., 2025). The component takes the original question, along with the reasoning and search queries from the trajectory, to critically assess why the current search trajectory is stuck. It then generates a new reasoning step and search query to help break out of the loop.

CONTROLLER: Figure 13 presents the controller prompt, which consists of signal definitions, action definitions, rules specifying when to choose each action, and additional guidelines. We intentionally avoid defining explicit thresholds or numerical values in the decision-making rules, allowing the LLM to interpret the signals by itself. All parts of this prompt were carefully designed through a precise prompt engineering process.

C Results

Intervene - Critical Thinking

You are a critical rethinking agent for an iterative deep research agent.
The research agent answers complex questions by repeatedly generating sub-queries, retrieving documents, and reasoning over the results.
A trajectory tracker monitors productivity signals (document novelty, query similarity) after each iteration. When those signals indicate the trajectory has stagnated — retrieving the same documents and repeating similar queries — the tracker triggers you to redirect the search.

Your task is to:
1. Critically assess why the current search trajectory is stuck.
2. Generate a creative, fundamentally different search query to break out of the loop.

== Inputs ==
- Original question: The user's top-level research question. Your new query must remain relevant to this question.
- Last N turns: The most recent search iterations, each containing:
* - Think: The agent's reasoning about what to search next.*
* - Search query: The sub-query the agent issued to the retriever.*

== Output Format ==
You MUST respond with ONLY a valid JSON object (no markdown code blocks, no ```json tags).
The JSON object must have exactly these two keys:

1. "reasoning":
* - Explain why the search trajectory is stuck*
* - Identify which query patterns are being repeated and why they fail to retrieve new relevant documents*
* - Suggest a fundamentally different angle or perspective to explore*

2. "search_query" (a new search query):
* - A creative, fundamentally different search query to redirect the trajectory*
* - Must target an unexplored aspect of the original question and use substantially different terminology*
* - Must be concise, keyword-based, and specific*

Example structure:
{
* "reasoning": "your reasoning here",*
* "search_query": "your new search query here"*
}

CRITICAL: Return ONLY the raw JSON object. Do NOT wrap it in markdown code blocks.

Figure 12: Intervene (Critical rethinking) Prompt.

C.1 RQ2: Retrieved Documents Distribution

Section 3.3 discusses the distribution of document types per iteration. In this section, we analyze these patterns from a cumulative perspective. Figure 14 shows the cumulative counts of three types of distractor documents—Relevant Seen, Irrelevant New, and Irrelevant Seen—across iterations. The plots indicate that as the number of iterations increases, the number of irrelevant and repeated documents in the trajectory also increases. This suggests that DRAs are unable to sufficiently diversify their search queries or generate optimal queries that retrieve relevant new documents. Additionally, they struggle to detect stagnation in the search process and therefore fail to stop iterating at the appropriate time.

C.2 RQ3 and RQ4: Controller Performance

To complement Tables 1 and 2, Table 5 reports the performance of DRAs both with and without the controller in terms of *Recall@N*, *Precision@N*, *F1@N*, accuracy (for BROWSECOMPPLUS), N, and the number of search calls. We observe that the best-performing controller variants—N-C-D for BROWSECOMPPLUS and N-D for NEUCLIR and LEGALSEARCH—improve Precision in the majority of cases, which consequently leads to higher F1 scores. This indicates that, in addition to improving result completeness, the controller makes the agents more selective and reduces the number of distractor documents. For other controller configurations (i.e., those using different input signals), Precision is still higher in the majority of settings than the without controller setup.

In another analysis, Figure 15 illustrates the performance of DRAs with and without the controller across iterations. The vertical lines indicate the average number of iterations. We observe that, in most cases, recall reaches higher levels with fewer iterations when the controller is used. Moreover, the average number of iterations decreases, suggesting that the controller improves both effectiveness and efficiency from a per-iteration perspective.

Finally, Figure 16 shows the distribution of the number of gold documents retrieved across agents. Interestingly, approximately 40% and 35% of gold documents in NEUCLIR and LEGALSEARCH, respectively, cannot be retrieved by any of the seven agents. This number is around 20% for BROWSECOMPPLUS. This observation suggests that certain query types are particularly challenging to capture, highlighting an important limitation that future work should address across all current agents.

Controller

You are a trajectory decision maker for an iterative deep research agent. After each search iteration you receive productivity signals, signal history, and action history. Select one action to guide the agent's next step.

== Inputs ==
- Current iteration number
- Current signals: doc_novelty, consec_query_sim, aspect_coverage, orig_query_sim
- Recent signal history
- Recent action history

== Signal Definitions ==
Signals are listed in priority order. When signals conflict, follow priority.

1. doc_novelty [PRIMARY] (value in [0, 1]):
Fraction of retrieved documents not seen before.
High -> fresh information being discovered. Low -> recycling old documents.
This is the most important signal. Decisions should be driven primarily by doc_novelty and its trend over recent iterations.

2. consec_query_sim [SUPPORTIVE] (value in [0, 1]):
Cosine similarity between the current and previous query embeddings.
High -> near-identical queries (repetition). Low -> diverse queries.
Use this to confirm or reinforce patterns observed in doc_novelty. High consec_query_sim alongside dropping doc_novelty is a strong stagnation signal; on its own it is less decisive.

3. aspect_coverage [CONFIRMATORY] (structured summary):
Tracks how many information-need aspects of the original query have been addressed.
Reports: covered/total aspects, partial/total, not_covered/total, a list of critical gaps (aspects with no evidence), and a list of minor gaps (aspects with partial evidence).
Use this to modulate stop/intervene decisions already indicated by doc_novelty. All aspects (partially or fully) covered confirms stop; not_covered aspects favor intervene over stop. When doc_novelty is moderate-to-high, ignore this signal.

4. orig_query_sim [GUARDRAIL] (value in [0, 1]):
Cosine similarity between the current query embedding and the original user question.
High -> on-topic. Low -> drifted from the original question.
This signal does not drive continue/intervene/stop decisions. Its sole role is as a safety check: only act on it when it drops very low, indicating the agent is generating unrelated subqueries and retrieving irrelevant documents.

== Action Definitions ==
"continue" — The trajectory is productive; no intervention needed.
Choose when:
- [Primary] doc_novelty is moderate-to-high, indicating fresh documents are still being found.
- [Supportive] consec_query_sim is low-to-moderate, confirming query diversity.
- [Confirmatory] aspect_coverage does not affect this decision. When doc_novelty is healthy, continue regardless of coverage status.
- [Guardrail] orig_query_sim is not alarmingly low (the agent has not drifted into irrelevant territory).

"intervene" — The trajectory is stale but recoverable. An external nudge (e.g., critical thinking) can redirect the search.
Choose when:
- [Primary] doc_novelty has dropped noticeably and sustained a decline over recent iterations. This is the main trigger for intervention.
- [Supportive] consec_query_sim is high, reinforcing the stagnation pattern (query loop).
- [Confirmatory] Not_covered aspects alongside near-zero doc_novelty favors intervene over stop. Key aspects of the question remain unaddressed, so stopping would produce an incomplete result.
- [Guardrail] A very low orig_query_sim alone can also trigger intervention if the agent has drifted far off-topic, even when doc_novelty appears adequate (novel but irrelevant documents).
- If a prior "intervene" was issued, enough iterations have passed to assess its effect.

"stop" — The trajectory is exhausted. Further iterations are unlikely to yield new information.
Choose when:
- [Primary] doc_novelty has been near zero over many consecutive iterations. This is the essential condition for stopping.
- [Supportive] consec_query_sim has been persistently high over the same stretch, confirming the agent is stuck.
- [Confirmatory] All aspects (partially or fully) covered alongside near-zero doc_novelty strengthens the case for stopping. The agent has addressed every facet and is out of new information.
- Multiple well-spaced interventions have failed to recover doc_novelty.

== Decision Guidelines ==
1. Default to "continue" when the trajectory is healthy. Expect natural signal fluctuation.
2. Weight signals by iteration stage: early iterations naturally have higher novelty and lower query similarity, so favor exploration and prefer "continue" unless doc_novelty is severely low. Later iterations tend to converge, making the same novelty drop more significant.
3. Respect the agent's self-correction ability. After issuing "intervene" or before resorting to "stop", wait several iterations — the agent may recover on its own. Avoid back-to-back interventions or premature stops.
4. Reserve "stop" for thoroughly demonstrated exhaustion: doc_novelty near zero over many iterations, multiple interventions attempted, each given time, none recovered doc_novelty. Full aspect coverage confirms stop; uncovered aspects favor intervene over stop. Do not escalate to "stop" without a sufficient observation window after the last intervention.

== Output Format ==
Respond with ONLY a raw JSON object — no markdown, no code blocks.
Keys:
- "reasoning": Brief explanation referencing current signals, recent history, and action history.
- "action": One of "continue", "intervene", "stop".
Example: {"reasoning": "your reasoning here", "action": "your selected action here"}

Figure 13: Controller Prompt.

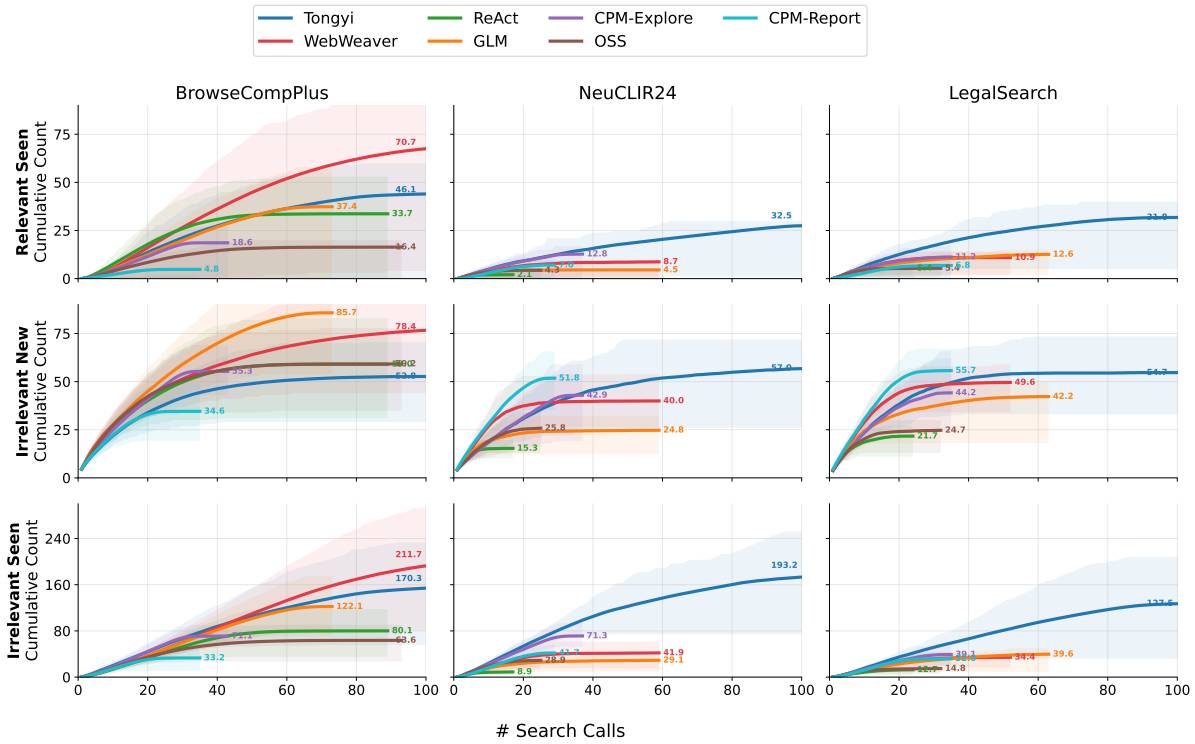

Figure 14: The distribution of document types across seven agents and three datasets highlights repetitive and low-diversity retrieval behavior in DRAs. Shaded bands show the variability (standard deviation) across queries.

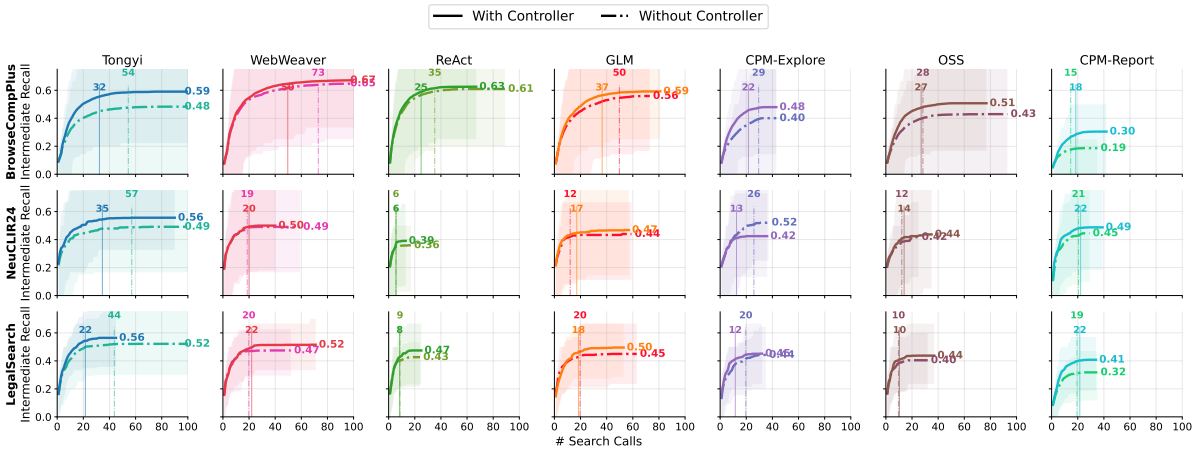

Figure 15: Intermediate recall (R_t) across seven agents and three datasets, with and without the controller, across iterations. Shaded bands indicate the standard deviation across queries. Vertical lines indicate the average number of iterations across queries.

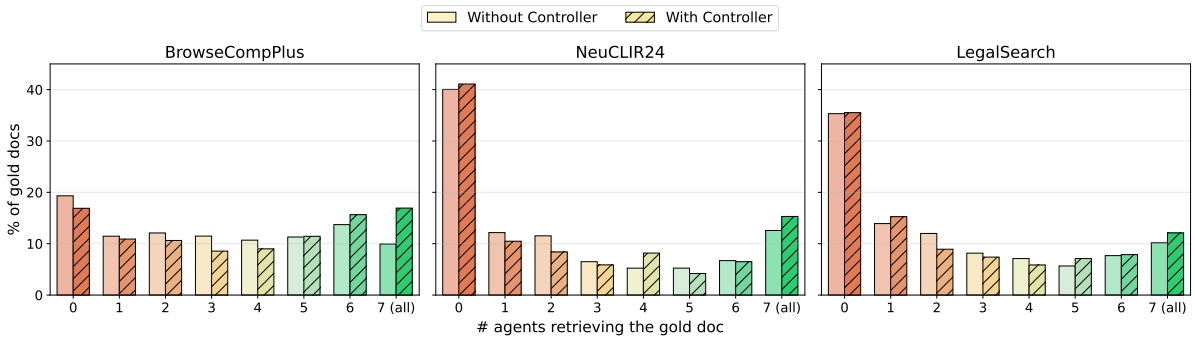

Figure 16: Distribution of the number of agents that retrieved the gold document.

Agent	BrowseCompPlus						NeuCLIR					LegalSearch				
	R	P	F1	Acc	#S↓	N	R	P	F1	#S↓	N	R	P	F1	#S↓	N
CPM-Report	18.6	4.0	5.8	11.7	14.4	12.5	44.5	4.7	7.8	20.6	36.9	31.8	5.6	7.9	19.4	28.6
+ RAAC (N-C-D)	30.5	5.1	7.7	12.0	18.2	12.2	41.4	6.0	9.3	12.2	25.1	36.7	7.4	10.6	16.5	21.7
+ RAAC (N-D)	28.4	5.1	7.6	10.7	16.5	11.8	48.7	4.8	8.0	22.2	34.4	40.8	5.5	9.1	21.6	27.2
+ RAAC (C-D)	26.7	5.9	8.8	9.5	9.4	5.9	36.6	6.9	10.3	9.6	16.0	33.8	9.2	12.5	9.5	10.3
+ RAAC (N)	27.5	5.0	7.6	10.6	15.0	11.3	46.4	5.6	9.0	20.2	33.4	45.0	6.0	9.4	22.2	27.9
+ RAAC (D)	26.6	4.3	6.6	10.0	16.6	12.3	40.9	6.2	9.5	23.0	36.9	23.7	6.5	9.0	20.8	27.2
GPT-oss-20b	43.0	5.9	9.8	23.0	28.3	61.7	41.8	9.1	12.5	12.2	27.7	40.5	15.2	18.9	10.2	27.9
+ RAAC (N-C-D)	50.7	7.8	12.5	26.4	27.0	61.9	39.3	9.6	13.0	11.6	27.4	40.2	15.0	18.5	10.0	29.3
+ RAAC (N-D)	48.5	8.0	12.6	24.5	25.5	60.2	43.7	8.9	12.3	14.0	31.8	43.8	14.2	18.6	10.8	30.3
+ RAAC (C-D)	41.4	8.6	13.5	17.2	12.3	33.8	39.3	9.2	13.1	10.4	24.6	40.7	15.8	18.8	8.9	24.8
+ RAAC (N)	45.3	7.9	12.4	20.1	22.8	56.3	41.3	9.1	12.8	11.8	28.1	48.0	13.8	18.4	12.4	33.9
+ RAAC (D)	47.9	6.2	10.3	25.2	31.3	67.2	42.6	10.0	13.6	13.3	27.7	40.5	14.1	17.8	10.7	28.7
CPM-Explore	40.1	5.5	9.2	30.5	29.6	57.7	52.1	7.4	11.6	25.5	45.2	44.2	10.3	14.3	23.8	48.0
+ RAAC (N-C-D)	48.0	8.7	13.7	29.7	21.7	47.1	44.0	10.2	14.1	11.7	27.8	40.7	13.4	16.8	11.9	32.6
+ RAAC (N-D)	45.3	8.6	13.4	26.3	21.3	47.0	42.5	10.3	14.2	12.5	29.1	45.2	14.2	18.2	12.2	32.5
+ RAAC (C-D)	39.9	9.2	14.1	21.5	11.9	30.1	40.6	9.8	13.4	10.3	24.3	39.0	15.7	18.6	9.5	23.5
+ RAAC (N)	44.4	8.7	13.5	27.7	19.5	45.8	40.7	9.5	12.9	12.2	29.6	40.7	14.1	17.7	10.8	28.8
+ RAAC (D)	44.8	6.7	11.0	34.5	27.5	53.5	46.3	7.7	11.6	20.9	38.6	45.9	12.4	16.0	19.5	40.4
GLM-4.7	55.9	5.9	10.1	32.0	50.3	89.2	43.9	10.6	14.3	12.1	26.7	45.0	11.7	16.1	19.8	45.9
+ RAAC (N-C-D)	59.2	7.8	12.8	34.9	36.8	75.7	44.1	9.1	13.3	12.0	31.1	47.2	13.7	18.0	13.9	38.9
+ RAAC (N-D)	56.3	7.8	12.7	33.9	35.3	74.7	46.8	8.9	13.0	17.1	42.1	50.0	11.1	15.9	18.6	48.5
+ RAAC (C-D)	45.1	8.5	13.6	26.9	14.7	37.6	41.1	9.5	13.4	10.0	25.5	44.4	14.3	18.2	11.2	31.6
+ RAAC (N)	53.6	8.7	13.8	32.7	28.5	64.9	42.5	8.8	12.7	13.6	33.6	50.4	12.0	16.9	18.9	49.6
+ RAAC (D)	60.5	5.8	10.1	35.7	52.9	91.4	41.0	8.9	12.5	14.5	30.4	48.1	12.8	17.2	20.5	46.7
ReAct	60.9	10.2	16.2	35.7	35.3	62.7	35.9	11.2	14.8	5.6	16.9	42.6	19.0	22.4	8.7	25.3
+ RAAC (N-C-D)	62.6	11.4	17.8	39.1	25.4	55.8	38.4	11.7	15.7	5.4	17.3	43.5	18.8	22.4	7.8	24.3
+ RAAC (N-D)	61.7	11.9	18.4	38.9	24.0	54.6	39.1	11.4	15.2	5.7	17.6	47.5	17.8	21.6	8.4	25.6
+ RAAC (C-D)	52.6	11.8	18.1	31.9	14.0	35.1	37.0	12.4	15.7	4.1	15.2	42.2	18.1	20.8	6.7	23.1
+ RAAC (N)	58.2	11.2	18.4	34.4	21.1	50.7	38.5	11.2	14.8	5.7	18.0	42.2	16.6	20.2	8.4	25.9
+ RAAC (D)	61.8	10.3	16.4	36.1	32.6	61.1	37.6	12.3	15.8	5.4	16.1	43.8	17.5	21.3	9.5	26.7
WebWeaver	64.9	7.7	13.0	47.8	75.0	11.8	49.0	7.0	11.0	19.2	30.5	47.5	9.2	14.1	19.9	21.7
+ RAAC (N-C-D)	67.2	8.4	14.1	51.3	50.7	15.4	47.9	7.0	10.8	19.8	29.0	50.1	9.5	14.3	19.1	15.2
+ RAAC (N-D)	65.7	8.7	14.5	51.8	42.1	15.5	50.0	6.5	10.5	20.3	31.1	51.6	8.4	13.3	22.1	22.6
+ RAAC (C-D)	56.7	8.5	14.2	43.2	25.6	13.0	46.5	7.6	11.7	16.9	25.3	46.2	10.3	14.9	16.4	13.8
+ RAAC (N)	62.0	8.8	14.7	46.8	33.7	15.0	48.0	6.5	10.2	20.0	31.0	43.3	15.6	19.3	22.0	24.2
+ RAAC (D)	67.7	7.5	12.9	51.2	74.0	14.0	44.9	8.1	11.9	21.4	31.9	44.6	15.6	21.0	19.8	24.6
Tongyi-DR	48.4	7.2	11.6	37.9	77.8	55.7	49.1	6.8	10.5	62.1	59.6	52.2	9.0	13.7	53.8	59.3
+ RAAC (N-C-D)	59.0	9.1	14.7	47.9	33.6	59.7	47.9	8.5	12.2	21.1	47.0	49.7	13.0	17.6	14.4	41.9
+ RAAC (N-D)	56.9	9.4	14.8	49.1	31.3	56.8	55.6	7.8	11.7	34.4	66.4	56.5	10.0	15.1	22.8	53.9
+ RAAC (C-D)	48.1	9.2	14.6	45.6	18.1	37.1	47.8	9.5	14.0	16.9	33.0	43.6	13.5	17.7	12.1	30.5
+ RAAC (N)	53.7	9.5	14.9	48.3	25.7	49.5	51.0	9.0	12.9	29.2	58.8	55.9	11.0	16.3	24.4	56.6
+ RAAC (D)	61.2	7.4	12.4	46.3	57.2	69.7	55.9	6.1	10.0	58.9	80.6	54.0	8.8	13.6	40.5	66.3

Table 5: Performance of DRAs in the original setting and with the controller. **R**: recall, **P**: precision, **F1**: F1, **Acc**: accuracy, **#S**: number of search calls, and **N**: number of seen documents.