Learning to Retrieve from Agent Trajectories

Yuqi Zhou[*1], **Sunhao Dai**[*†1], **Changle Qu**[1], **Liang Pang**[2], **Jun Xu**[✉1] **and Ji-Rong Wen**[1]
[1]Gaoling School of Artificial Intelligence, Renmin University of China, [2]CAS Key Laboratory of AI Safety, Institute of Computing Technology, Chinese Academy of Sciences

Information retrieval (IR) systems have traditionally been designed and trained for human users, with learning-to-rank methods relying heavily on large-scale human interaction logs such as clicks and dwell time. With the rapid emergence of large language model (LLM) powered search agents, however, retrieval is increasingly consumed by agents rather than human beings, and is embedded as a core component within multi-turn reasoning and action loops. In this setting, retrieval models trained under human-centric assumptions exhibit a fundamental mismatch with the way agents issue queries and consume results. In this work, we argue that retrieval models for agentic search should be trained directly from agent interaction data. We introduce *learning to retrieve from agent trajectories* as a new training paradigm, where supervision is derived from multi-step agent interactions. Through a systematic analysis of search agent trajectories, we identify key behavioral signals that reveal document utility, including browsing actions, unbrowsed rejections, and post-browse reasoning traces. Guided by these insights, we propose LRAT, a simple yet effective framework that mines high-quality retrieval supervision from agent trajectories and incorporates relevance intensity through weighted optimization. Extensive experiments on both in-domain and out-of-domain deep research benchmarks demonstrate that retrievers trained with LRAT consistently improve evidence recall, end-to-end task success, and execution efficiency across diverse agent architectures and scales. Our results highlight agent trajectories as a practical and scalable supervision source, pointing to a promising direction for retrieval in the era of agentic search.

GitHub: https://github.com/Yuqi-Zhou/LRAT
Homepage: https://yuqi-zhou.github.io/LRAT-homepage/
Collection: https://huggingface.co/collections/Yuqi-Zhou/lrat

[*]Equal contribution. [†]Project Leader. [✉]Corresponding author.

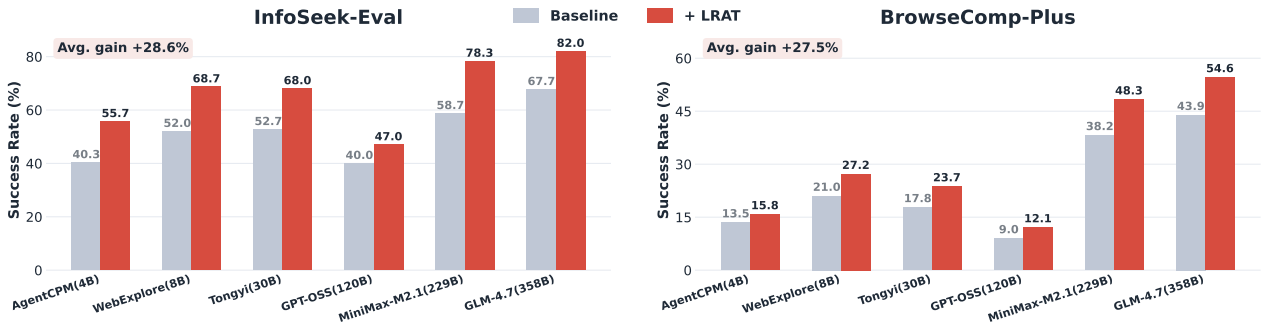


Figure 1 | Overview of LRAT gains across six agent backbones with Qwen3-Embedding-0.6B as the retriever. Left: Success rate on InfoSeek-Eval. Right: Evidence recall on BrowseComp-Plus. LRAT consistently improves both in-domain task success and out-of-domain retrieval quality.

Emails: yuqizhou@ruc.edu.cn, sunhaodai@ruc.edu.cn, junxu@ruc.edu.cn

1. Introduction

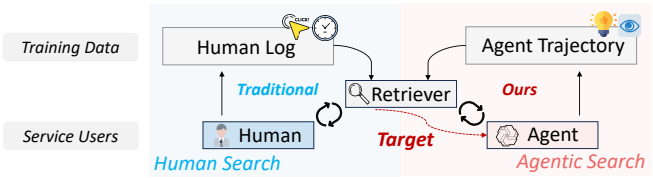


Figure 2 | Illustration of the search paradigm shift where the retriever's target changes from humans to agents.

Information retrieval (IR) has long served as the foundation of information access systems such as web search engines Baeza-Yates et al. (1999); Chowdhury (2010); Croft et al. (2010); Manning (2009), and decades of research in learning to retrieve and rank have been built around a human-centric paradigm Cao et al. (2006); Liu et al. (2009); Xu and Li (2007). In this setting, retrieval models are trained from large-scale human interaction logs (e.g., clicks Joachims (2002); Joachims et al. (2005) and dwell time Kelly and Belkin (2004); Kim et al. (2014)) and optimized to serve humans Joachims (2002); Joachims et al. (2005); Shen et al. (2005), forming a powerful data flywheel. With the rapid emergence of large language model (LLM) powered agents, however, this paradigm is being fundamentally challenged Huang et al. (2025); Shi et al. (2025); Xu and Peng (2025). Search engines are increasingly queried by agents rather than humans, and retrieval is no longer a standalone endpoint but a core tool embedded within an agent's multi-turn reasoning and action loop Dai et al. (2025); White (2024); Xi et al. (2025). Search agents iteratively issue sub-queries, consume retrieved information, and refine their actions to solve complex tasks, making retrieval quality a critical bottleneck that directly constrains what information agents can observe, reason over, and ultimately accomplish.

Despite this shift in usage, today's search agents typically rely on general-purpose retrieval models Jin et al. (2025); Song et al. (2025); Wang et al. (2025), such as dense embedding retrievers, or external search APIs (e.g., Google or Bing). As shown in Figure 2, these retrievers are overwhelmingly trained from human interaction logs and implicitly encode human-centric assumptions about how queries are issued, how results are examined, and how relevance signals are expressed. When the primary user becomes an agent, these assumptions break in fundamental ways. Agent queries are not issued to satisfy immediate informational needs, but to advance intermediate reasoning objectives during problem-solving, resulting in relevance patterns that differ from those of human users. As a result, search agents are still served by retrieval models that are trained primarily from human data, creating a fundamental mismatch between how retrieval models are trained and how they are used in agentic search.

This mismatch motivates a rethinking of retrieval training for the agent era. In this work, we argue that retrieval models should be trained directly from agent interaction data, rather than repurposed from human-centric search. Analogous to how learning to retrieve and rank from user interaction logs in web search, we propose **learning to retrieve from agent trajectories** as a new training paradigm. Agent trajectories record the sequence of intermediate queries, retrieved documents, and reasoning steps generated during task execution, providing a rich and naturally abundant source of supervision. Training retrievers on such trajectories intrinsically aligns retrieval objectives with agents' feedback. Moreover, because agent trajectories are generated as a byproduct of every agent invocation, they create an opportunity to build a sustainable data flywheel for retrieval in agentic search, positioning trajectories as the agent-era counterpart of user click logs.

In this paper, we take an early step toward this direction by systematically analyzing how deep research agents interact with retrieval systems through multi-turn execution trajectories. Our analysis reveals fundamental differences between human feedback and agent feedback, and identifies key agent behaviors that reliably indicate document utility. Specifically, we show that (1) browsing actions constitute a necessary condition for task success, making browsed documents

natural candidates for naive positive signals; (2) unbrowsed documents in agent trajectories serve as reliable negatives without position bias; and (3) post-browse reasoning traces provide a strong indicator of relevance intensity, distinguishing truly useful documents from superficially browsed ones. Guided by these insights, we propose **LRAT** (**L**earning to **R**etrieve from **A**gent **T**rajectories), a simple yet effective framework for training retrieval models directly from agent interactions. LRAT first mines coarse query-document supervision from search-browse transitions and refines positive signals using explicit post-browse reasoning traces. It then incorporates relevance intensity through reasoning-length–aware weighting to prioritize documents that drive substantial agent progress. In our instantiation, LRAT is applied to trajectories generated by Tongyi-DeepResearch-30B on 10K InfoSeekQA queries with four retrievers, yielding **26,482** agent trajectories and **91,713** training pairs. Importantly, LRAT requires no additional human annotation and can be applied to trajectories generated by arbitrary agents and retrievers, making it a practical and scalable approach for agent-centric retrieval training.

In summary, our main contributions are threefold:

• We identify a fundamental misalignment between human-centric retrieval training and agentic search, and formulate *learning to retrieve from agent trajectories* as a new retrieval paradigm. In this setting, supervision is derived from multi-step agent interactions, reflecting how retrieval is actually used by search agents.

• Guided by insights from empirical analysis, we propose LRAT, a simple yet effective framework that mines high-quality retrieval supervision from agent trajectories, providing a practical step toward agent-aligned retriever training.

• Experiments on both in-domain and out-of-domain deep research benchmarks show that LRAT consistently improves evidence retrieval and end-to-end agent performance across diverse agent architectures and scales. We further demonstrate that LRAT can support a self-improving data flywheel, highlighting the scalability value of LRAT in real-world scenarios.

2. Related Work

Search Agent & Retriever Optimization for RAG. The rise of agentic search, exemplified by deep search agents such as Search-o1 Li et al. (2025) and Search-R1 Jin et al. (2025), has revolutionized how LLMs handle complex information-seeking tasks through multi-step reasoning and iterative interactions with search engines. While significant progress has been made in optimizing the agent Team et al. (2025b); Wang et al. (2025); Zheng et al. (2025), the underlying retrieval model has largely been treated as a static, off-the-shelf tool (e.g., Google Search API, BM25, or dense embedding models). This overlooks a fundamental mismatch between static retrievers and the interactive, multi-step nature of agentic search, causing retrieval to emerge as the primary performance bottleneck for search agents. Although traditional Retrieval-Augmented Generation (RAG) research has explored retriever optimization through preference alignment Dong et al. (2025); Qu et al. (2025); Shi et al. (2024), these methods are primarily designed for single-turn retrieval based on a user's initial, explicit query. In contrast, agentic search operates in a multi-step and interactive manner. Search queries are not the fixed initial user query but intermediate actions generated by the agent during reasoning. These queries are highly dynamic and cannot be judged as correct based on a final answer, unlike prior work Ke et al. (2024); Salemi and Zamani (2024); Zamani and Bendersky (2024); Zhang et al. (2024). In contrast to prior work, our work shifts the focus from merely refining the agent to optimizing the retriever, enabling it to respond effectively to the agents' information needs as they evolve across multiple turns.

Learning to Retrieve/Rank from Human Feedback. Learning to retrieve/rank has a long history in the information retrieval community Cao et al. (2006, 2007); Liu et al. (2009); Xu and Li (2007). Early work relies on explicit relevance judgments annotated by human assessors, forming the foundation of supervised learning-to-rank methods that optimize document ordering using pointwise, pairwise, or listwise objectives Burges et al. (2005); Burges (2010); Li et al. (2007). To overcome the scalability bottlenecks of manual annotation, a significant body of research shifted toward implicit user feedback, leveraging interaction signals such as clicks Joachims (2002); Joachims et al. (2005), dwell time Kelly and Belkin (2004); Kim et al. (2014), and scrolling patterns Agichtein et al. (2006a,b); Radlinski et al. (2008). These methods treat user interaction logs as weak supervision signals, employing click models or counterfactual learning to refine ranking models at scale. More recently, the primary users of modern search systems are no longer limited to humans, but increasingly include autonomous agents that interact with search engines. Motivated by this shift, our work investigates learning to retrieve from agent trajectories, where supervision is derived from the agent's sequential interactions with the search system.

3. Preliminaries

3.1. Deep Research Agent Trajectories

In this paper, we focus on *Deep Research Agents*, i.e., LLM-powered intelligent agents that solve complex information-seeking tasks via iterative interactions with an external retrieval system. A typical execution trajectory of a Deep Research Agent is illustrated in Figure 3.

Formally, given an initial user query q, a Deep Research Agent follows a ReAct-style Yao et al. (2022) interaction pattern, repeatedly alternating between *thinking* and *acting*, which yields a multi-turn execution trajectory $\mathcal{T} = \{(r_t, a_t, o_t)\}_{t=1}^{T}$. At each turn t, the agent maintains an internal reasoning state r_t that guides subsequent actions a_t, incorporates new observed information o_t returned from the external retrieval system, and determines whether further information is required.

[Think]: At the beginning of each turn t, the agent first produces an explicit reasoning state (or *thought*) r_t. This reasoning step analyzes the current context, summarizes previously acquired evidence, and identifies what information is still missing for answering the original user query.

[Search] or [Browse]: Conditioned on this reasoning state, the agent selects an action $a_t \in \mathcal{A}$ to interact with the external retrieval environment. We consider two typical actions:

• When invoking the **[Search]** action, the agent generates an intermediate search query q_t that targets a specific information gap identified in the reasoning state r_t. In response, the retrieval system returns a ranked set of top-K candidate documents $\mathcal{D}_t = \{d_{t,i}\}_{i=1}^{K}$. In practice, the agent does not immediately observe the full content of each document. Instead, it receives **a snippet list** for $\mathcal{D}_t$ (e.g., titles and brief summaries) as observation o_t, which provides coarse-grained signals for initial assessment and subsequent decision making.

• When invoking the **[Browse]** action, the agent selects one document d_t from the previously retrieved candidates $\mathcal{D}_{t'}(t' < t)$ and requests to read it in full. The retrieval system then returns the **complete content** of the selected document as the observed information o_t for this turn. The agent incorporates this information into its reasoning state and continues the cycle of reasoning and action across multiple turns.

[Answer]: This iterative process continues until the agent determines that sufficient information has been gathered to answer the original query. At termination, the agent performs a final reasoning step that synthesizes the accumulated evidence across the trajectory and generates the final answer, which is returned to the user.

Trajectory Example

User Query: Which station merged with Immaculate Heart Radio in 2017?

Agent Trajectory

[Think]: No merger info yet. Let's search for the 2017 merger details.

[Search]: `Search["Immaculate Heart" "merged" "2017"]`

[Information]: Found 10 results:
Doc [3948]: Immaculate Heart Radio ...
Doc [2295]: corporate name Relevant Radio, Inc...
...

[Think]: Doc [2295] mentioned Relevant Radio. It might contain the specific merger date. Let's read it.

[Browse]: `Browse[DocID: 2295]`

[Information]: ... Relevant Radio announced it would merge...

... Intermediate turns omitted ...

[Think]: I have found the answer...

[Answer]: The station is WMJR (Nicholasville, KY).

Figure 3 | An example of Deep Research Agent trajectory.

3.2. Task Definition

Most existing search agents employ off-the-shelf retrieval models, such as pretrained dense retrievers (e.g., Qwen3-Embedding or E5-Embedding) or external search APIs (e.g., Google or Bing). These retrievers are typically trained from human interaction logs and optimized for human-centric search behavior. In contrast, deep research agents generate rich multi-turn execution trajectories that reveal how retrieval results are examined and utilized. Despite being naturally abundant, such trajectory data is largely untapped for training retrieval models. This raises a natural question: *can retriever be trained directly from agent trajectories?*

We define the task of *Learning to Retrieve from Agent Trajectories* as follows: given a collection of agent execution trajectories $\{\mathcal{T}\}$ as defined in Section 3.1, the goal is to learn a retrieval model that produces ranked document lists optimized to support an agent's multi-step reasoning and problem-solving process. Unlike traditional learning-to-rank, supervision in this task should be directly derived from agent trajectories, aligning retrieval training directly with agent behaviors.

4. Analysis of Agent Trajectories

In this section, we conduct a systematic analysis of deep research agent trajectories. Our primary goal is to understand how agents interact with retrieval systems in practice, identify patterns that differ from human search, and extract insights that inform the design of agent-centric relevance modeling.

4.1. Environment Setup

This section describes the experimental setup used to generate and analyze deep research agent trajectories.

Table 1 | Statistics of generated trajectories across different retrievers. For each retriever, we report the number of completed trajectories (N) and the average numbers of [Search] actions (Avg. S), [Browse] actions (Avg. B), their ratio (B/S), and the total execution steps (Avg. T), categorized into correct, incorrect, and all completed trajectories.

Retriever	**Correct**					**Incorrect**					**Total (Complete)**				
	N	**Avg.** S	**Avg.** B	B/S	**Avg.** T	N	**Avg.** S	**Avg.** B	B/S	**Avg.** T	N	**Avg.** S	**Avg.** B	B/S	**Avg.** T
BM25	7,674	9.15	2.96	0.32	12.11	1,872	29.15	5.97	0.20	35.11	9,546	13.07	3.55	0.27	16.63
Qwen3-Embedding-0.6B	5,913	12.81	3.68	0.29	16.49	2,062	38.95	7.17	0.18	46.12	7,975	19.57	4.58	0.23	24.15
Qwen3-Embedding-4B	6,354	13.24	4.11	0.31	17.34	2,121	36.13	7.47	0.21	43.60	8,475	18.97	4.95	0.26	23.91
Qwen3-Embedding-8B	6,541	11.86	3.69	0.31	15.55	2,082	34.47	7.20	0.21	41.67	8,623	17.32	4.54	0.26	21.85
Total	**26,482**	11.77	3.61	0.31	15.38	**8,137**	34.68	6.95	0.20	41.63	**34,619**	17.25	4.41	0.26	21.66

4.1.1. Seed Data Selection

To encourage sustained search and browsing behavior, we adopt InfoSeekQA Xia et al. (2025) as the seed dataset for trajectory generation. InfoSeekQA is a large-scale deep research benchmark comprising over 50K question–answer pairs, designed to require hierarchical reasoning and iterative information acquisition. Tasks in InfoSeekQA typically involve substantially deeper search processes, resulting in significantly longer interaction trajectories than traditional QA datasets. Given computational constraints and evaluation reliability, we select the top 10K queries with verified ground-truth answers as the seed set for trajectory construction. Following the standard practice, we use the Wiki-25-Dump[1] as the corpus, which comprises over 11.2 million document chunks, each truncated to 512 tokens.

4.1.2. Retrieval Systems

To cover a diverse range of retrieval behaviors, we deploy four retrieval models: a sparse *BM25* retriever Robertson et al. (2009) and three dense retrievers of increasing capacity, *Qwen3-Embedding-0.6B, 4B, and 8B* Yang et al. (2025), spanning lexical matching and semantic retrieval capabilities.

During trajectory generation, each [Search] action returns the top-10 candidate documents. For each candidate, the agent observes a short snippet consisting of the first 64 tokens of the document, which approximates the average length of web search snippets measured under the Qwen3 tokenizer. This design simulates a realistic search environment where agents initially rely on coarse-grained evidence before deciding whether to browse full documents.

4.1.3. Deep Research Agent Configuration

We adopt *Tongyi-DeepResearch-30B-A3B*[2] Team et al. (2025b) as the search agent for trajectory collection, which is one of the strongest open-source search agents tailored for long-horizon, deep information-seeking tasks and supports over one hundred interaction steps. To allow sufficient exploration, we set the maximum number of interaction rounds to $T = 100$. If the agent fails to reach a conclusion within this limit, it is required to produce a final answer based on the information collected so far. A trajectory is considered valid only if the final answer matches the ground truth provided by InfoSeekQA.

[1] https://huggingface.co/datasets/Lk123/wiki-25-512

[2] https://modelscope.cn/models/iic/Tongyi-DeepResearch-30B-A3B

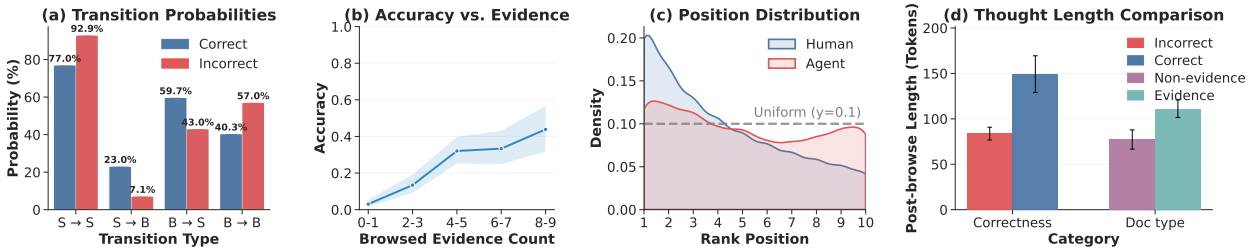


Figure 4 | Trajectory analysis on BrowseComp-Plus, where each user query is annotated with supporting evidence documents. Interaction traces are generated by *Tongyi-DeepResearcher* with the *Qwen3-Embedding-0.6B* retriever. Across all panels, *correct* and *incorrect* denote trajectories leading to correct and incorrect final answers. (a) Action transition probabilities between [Search] (S) and [Browse] (B). (b) Accuracy versus the number of unique evidence documents browsed (binned; shaded areas indicate 95% Wilson confidence intervals). (c) Browsing rank-position distribution under shuffled documents (agent vs. human Craswell et al. (2008)). (d) Length of post-browse reasoning (tokens) by trajectory correctness (correct vs. incorrect) and document type (evidence vs. non-evidence).

4.1.4. Trajectory Generation

Using the environment described above, we generate agent trajectories by executing the search agent on each seed query. For each question, the agent iteratively produces reasoning traces and executes [Search] or [Browse] actions until termination. Figure 3 illustrates a representative execution trajectory. We filter out trajectories that exceed the maximum step limit or produce incorrect final answers. Answer correctness is verified by comparing the agent's output against the ground truth using *Qwen3-30B-A3B-Thinking-2507*[3] Yang et al. (2025). The resulting trajectories form the basis of our analysis, and their statistics are summarized in Table 1.

4.2. Agent Trajectory Analysis

In this section, we analyze deep research agent trajectories to understand how agents interact with retrieval systems during long-horizon problem solving.

4.2.1. Browsing Is a Necessary Signal for Successful Retrieval

We first investigate which agent behaviors are most indicative of effective information acquisition. By comparing successful and failed trajectories, we observe a clear behavioral divergence in how agents interact with the retrieval system. As summarized in Table 1, failed (i.e., incorrect) trajectories exhibit a substantially lower ratio between [Browse] and [Search] actions (B/S). This pattern indicates that agents in unsuccessful runs frequently issue search queries but rarely proceed to browse retrieved documents, suggesting that the returned snippets are often deemed insufficiently informative to warrant further inspection. In contrast, successful (i.e., correct) trajectories show fewer repetitive search actions and a markedly higher frequency of browsing behaviors, reflecting more effective utilization of retrieved results.

This pattern is further confirmed by the action transition statistics shown in Figure 4(a). Successful trajectories show a significantly higher probability of transitioning from [Search] (S) to [Browse] (B), whereas unsuccessful trajectories are more likely to remain in search-only loops without progressing to document consumption. Moreover, Figure 4(b) shows that task success

[3] https://huggingface.co/Qwen/Qwen3-30B-A3B-Thinking-2507

increases monotonically with the number of browsed evidence documents, and drops to zero when the agent never browses any document containing the required evidence.

Together, these results indicate that browsing retrieved documents is not merely correlated with task success, but constitutes a necessary condition for successful task completion. This observation motivates **treating browsed documents as primary candidates for positive supervision** when training retrieval models for search agents.

4.2.2. Unbrowsed Documents Are Reliable Negatives

Having identified browsing as a strong indicator of positive utility, we next examine whether documents that are not browsed can be interpreted as negative signals in agent trajectories. In human-centric click logs, negative signals are notoriously ambiguous due to well-known position bias or exposure bias Joachims et al. (2005): unclicked documents may be irrelevant or simply unseen. As a result, learning-to-rank methods typically adopt conservative heuristics, such as "skip-above" sampling Granka et al. (2004); Joachims (2002); Joachims et al. (2005), to avoid introducing false negatives.

To assess whether similar issues arise in agent trajectories, we analyze the distribution of browsing actions across ranking positions. Figure 4(c) shows that, unlike human clicks, the agent's browsing behavior is not sharply concentrated at top ranks. Instead, browsing is distributed relatively evenly across positions, indicating that the agent actively evaluates candidates beyond the highest-ranked results rather than relying on positional cues. This suggests that unbrowsed documents are typically the result of explicit rejection after inspection, rather than limited exposure. Therefore, in contrast to human click logs, **all unbrowsed documents within a retrieved candidate set can be treated as reliable negatives** without requiring position bias correction.

4.2.3. Post-Browse Reasoning Traces Are Important Indicators

While browsing behavior identifies documents that the agent chooses to inspect, it remains an implicit signal and may include noise. Deep research agents, however, generate explicit reasoning traces immediately following a [Browse] action, providing a more direct view of how retrieved content is interpreted and utilized during problem solving.

We analyze the agent's post-browse reasoning to understand their relationship with task success and document utility. As shown in Figure 4(d), trajectories that ultimately produce correct answers are associated with significantly longer reasoning following browsing actions than those that lead to incorrect answers. A closer inspection of cases reveals a consistent pattern: in unsuccessful trajectories, the agent often quickly abandons a browsed document after determining that it does not contain useful information, resulting in short post-browse reasoning. In contrast, successful trajectories exhibit substantially longer reasoning after browsing, reflecting deeper analysis and integration of the retrieved content into subsequent decision making. Moreover, documents that contain ground-truth evidence are followed by markedly longer reasoning traces than non-evidence documents, indicating that useful content elicits more extensive agent reasoning.

Together, these observations suggest that **post-browse reasoning traces provide a reliable signal of document utility**. In particular, the length of the reasoning trace after browsing is strongly correlated with whether a document contributes meaningfully to task progress, offering valuable insight into more fine-grained relevance beyond binary feedback.

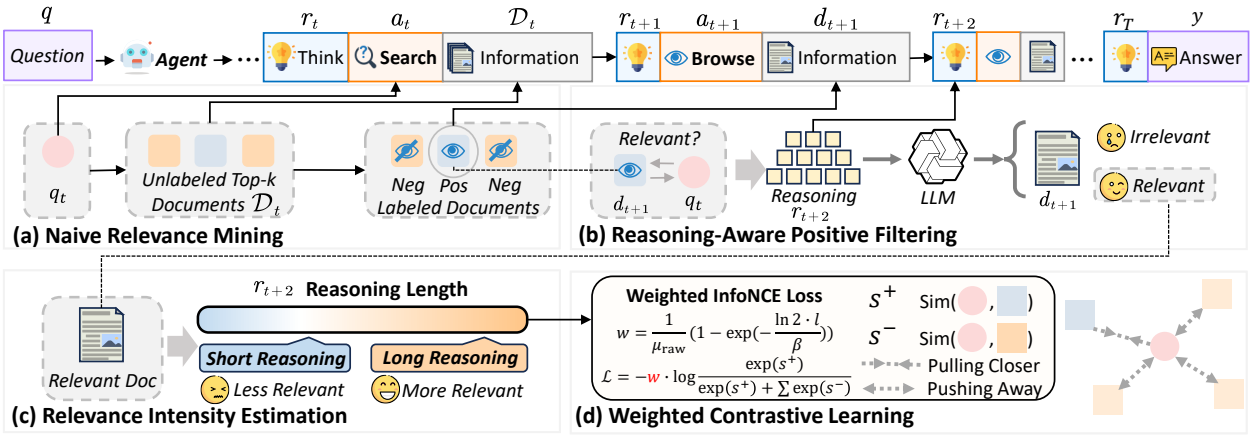


Figure 5 | Our proposed LRAT retrievers training framework using deep research agent interactions. (a) Extract relevance signals from *Browse* actions in the interaction sequence. (b) Filter out irrelevant documents by judging *post-Browse reasoning* with LLM. (c) Estimate relevance weights based on the length of *post-Browse reasoning*. (d) Perform contrastive learning using the filtered samples and their relevance weights.

5. Learning to Retrieve from Trajectories

Motivated by the insights in Section 4, we propose **LRAT**, a simple yet effective framework for training retrievers directly from deep research agent interactions. LRAT progressively mines relevance signals from trajectories to construct high-quality query-document supervision, and then optimizes a dense retriever with utility-aware weighting. An overview of the framework is shown in Figure 5.

5.1. Mining Relevance Signals

Given a collection of valid agent trajectories $\{\mathcal{T}\}$ (Section 4.1), our goal is to extract query-document supervision that reflects how agents actually consume and utilize retrieval results. Guided by the analysis in Section 4, we adopt a multi-stage, progressively refined relevance mining procedure. We start from coarse trajectory signals (browsing decisions) to obtain naive supervision, and then refine positives using post-browse reasoning traces to reduce noise.

5.1.1. Naive Relevance Mining from Search-Browse Transitions

We first construct coarse supervision from the agent's **[Search]** $\rightarrow$ **[Browse]** transitions. Consider a search turn t where the agent issues an intermediate query q_t and receives a ranked top-K candidate set $\mathcal{D}_t = \{d_{t,i}\}_{i=1}^{K}$. If the agent subsequently performs a [Browse] action on one of the candidates at the next turn, we view the browsed document d_{t+1} as a *naive positive* sample, since browsing is a necessary prerequisite for successful task completion (as analyzed in Section 4.2.1).

For negatives, human click logs require careful debiasing due to position bias. In contrast, our analysis in Section 4.2.2 shows that agent browsing decisions exhibit weak position dependence, suggesting that unbrowsed items within the retrieved candidate set are more likely to reflect explicit rejection rather than lack of exposure. Therefore, for each browsed document d_{t+1}, we treat all other candidates in the same retrieved set that are *not browsed* as naive negatives:

$$\mathcal{N}_t = \mathcal{D}_t \setminus \{d_{t+1}\}.$$

This yields coarse training instances of the form $(q_t, d_{t+1}, \mathcal{N}_t)$.

5.1.2. Reasoning-Aware Positive Filtering

Browsing actions are still imperfect proxies of relevance, because agents choose documents to browse based on coarse snippets and may later judge that a browsed document is unhelpful. Our analysis in Section 4.2.3 shows that post-browse reasoning traces provide a reliable indicator of document utility, often explicitly stating whether the browsed content resolves the information gap.

Based on this insight, we refine naive positives with a *reasoning-aware LLM-as-judge filter*. For each browsed document d_{t+1}, we collect the agent's immediate post-browse reasoning trace r_{t+2} and apply an LLM-based verifier to determine whether the reasoning explicitly uses the document content to support progress on the task. Concretely, we use *Qwen3-30B-A3B-Thinking-2507* as the judge and label (q_t, d_{t+1}) as `Relevant` or `Irrelevant` based on the reasoning trace r_{t+2}.

This filtering step removes clear noise among browsed-but-unhelpful documents while preserving high-quality positives. In our empirical validation on BrowseComp-Plus evidence annotations, the verifier can retain **97.2%** of ground-truth evidence documents, ensuring near-perfect recall of strong positives. Meanwhile, it retains **74.8%** of browsed non-evidence documents, indicating that the filter removes obvious noise while still capturing agent-specific utility that may go beyond rigid dataset evidence labels.

5.2. Intensity-Aware Training

We next describe how LRAT leverages the trajectory-derived supervision constructed in Section 5.1 to train a retrieval model. Beyond identifying which documents are relevant, agent trajectories also reveal *how strongly* a document contributes to task progress. LRAT explicitly incorporates this notion of relevance intensity into retriever optimization.

5.2.1. Reasoning-Length Induced Relevance Intensity Estimation

As analyzed in Section 4.2.3, post-browse reasoning provides a reliable indicator of document utility: longer reasoning chains following a browsing action are strongly correlated with higher document usefulness for the agent's subsequent planning and problem solving. This phenomenon is analogous to classical human search, where dwell time has long been recognized as an effective proxy for relevance intensity Kelly and Belkin (2004); Kim et al. (2014). In both cases, increased cognitive effort reflects deeper engagement with the retrieved content.

Motivated by this analogy, we introduce a relevance intensity estimation scheme tailored to agent-derived feedback. Rather than treating all positive signals equally, we model relevance as a continuous quantity derived from the agent's post-browse reasoning length. Inspired by the saturation principle underlying the time-aware click model Liu et al. (2016), we map reasoning length to a bounded utility score using an exponential saturation function, capturing the diminishing returns of increasingly long reasoning traces.

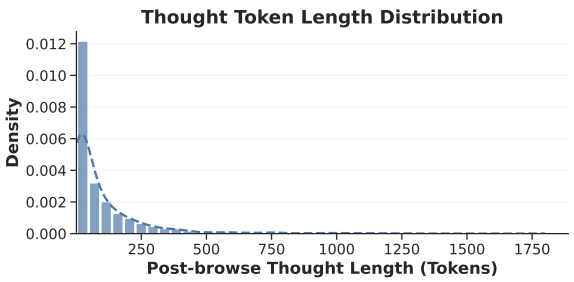


Figure 6 | Distribution of thought token lengths after browsing actions, computed from agent trajectories in the same setting as Figure 4.

In the time-aware click model Liu et al. (2016), the marginal gain at dwell length x follows an ex-

ponentially decaying function,

$$g(x) = \exp\left(-\frac{\ln 2}{\beta}x\right), \tag{1}$$

where β is the half-life of cumulative gain. Our analysis of post-browse reasoning lengths (Figure 6) shows a similar exponential decay in agent trajectories. Integrating this marginal gain from 0 to the observed reasoning length l yields the cumulative relevance utility:

$$u(l) = \int_0^l g(x)\,dx = \frac{\beta}{\ln 2}\left(1 - \exp\left(-\frac{\ln 2}{\beta}l\right)\right), \tag{2}$$

which naturally saturates for long reasoning traces. Normalizing across the dataset produces the relevance intensity weight used for training.

Formally, let $l = \text{Length}(r)$ denote the number of tokens in the reasoning trace immediately following the browsing of a document. We compute the relevance intensity weight as:

$$w = \frac{1}{\mu_{\text{raw}}}\left(1 - \exp\left(-\frac{\ln 2 \cdot l}{\beta}\right)\right), \tag{3}$$

where β is a length-scale parameter set to the median reasoning length across all trajectories, and μ_{raw} denotes the global mean of the unnormalized scores over the dataset. This normalization ensures $\mathbb{E}[w] \approx 1$, maintaining training stability while assigning higher importance to documents that trigger deeper agent reasoning and greater task progress. For simplicity, we omit the constant factor $\frac{\beta}{\ln 2}$, as the final weights are normalized across the dataset and the relative ranking of documents is preserved.

5.2.2. Weighted Contrastive Learning

We instantiate LRAT using a standard bi-encoder dense retriever, as commonly used in embedding models. Each query q and document d is independently encoded into vector representations $\mathbf{e}_q, \mathbf{e}_d \in \mathbb{R}^h$, where h denotes the embedding dimension. The relevance score is then computed via a similarity function, such as dot product or cosine similarity: $s(q,d) = \text{sim}(\mathbf{e}_q, \mathbf{e}_d)$.

To incorporate relevance intensity into optimization, we modify the standard InfoNCE loss Gutmann and Hyvärinen (2010) by introducing sample-wise weighting to train the dense retriever. For a mini-batch of size N, the weighted contrastive objective is defined as

$$\mathcal{L} = -\frac{1}{N}\sum_{i=1}^{N} w_i \cdot \log \frac{\exp\left(s(q_i, d_i^+)/\tau\right)}{\exp\left(s(q_i, d_i^+)/\tau\right) + \sum_{d^- \in \mathcal{N}_i} \exp\left(s(q_i, d^-)/\tau\right)},$$

where τ is the temperature. d_i^+ denotes the positive document for query q_i, and w_i is the reasoning-length–derived weight defined in Eq. (3). The weight w_i scales the gradient contribution of each instance, allowing documents associated with deeper reasoning to exert a stronger influence during training.

The negative set $\mathcal{N}_i$ is constructed from two complementary sources: (1) unbrowsed documents from the same retrieved candidate set, derived from agent trajectories as described in Section 5.1.1, and (2) in-batch negatives consisting of documents paired with other queries in the same mini-batch. This hybrid negative sampling strategy improves discriminative power, helping the retriever separate high-utility evidence from explicitly rejected candidates and unrelated documents, while avoiding representation collapse.

Table 2 | Results on task-optimized search agents and generalist agentic foundation models with different retrievers on in-domain (ID) InfoSeek-Eval and out-of-domain (OOD) BrowseComp-Plus benchmarks. Qwen3-Emb denotes Qwen3-Embedding-0.6B and E5-Large denotes Multilingual-E5-Large-Instruct. Metrics include Success Rate (SR), Recall, and Average Step Count (Avg. Steps). Best results within each agent backbone are highlighted in **bold**.

Agent Backbone	Retriever	InfoSeek-Eval (ID)		BrowseComp-Plus (OOD)		
		SR (↑)	Avg. Steps (↓)	SR (↑)	Recall (↑)	Avg. Steps (↓)
I. Task-Optimized Search Agents						
AgentCPM-Explore (4B)	Qwen3-Emb	40.3	38.0	13.5	23.2	40.7
	+ LRAT (Ours)	**55.7** (+38.2%)	**34.4**	**15.8** (+17.0%)	**32.0** (+37.9%)	**40.4**
	E5-Large	47.3	38.9	15.9	26.5	40.7
	+ LRAT (Ours)	**49.7** (+5.1%)	**35.5**	**15.9** (+0.0%)	**32.1** (+21.1%)	**40.1**
WebExplore (8B)	Qwen3-Emb	52.0	24.1	21.0	47.7	40.7
	+ LRAT (Ours)	**68.7** (+32.1%)	**19.0**	**27.2** (+29.5%)	**55.9** (+17.2%)	**38.7**
	E5-Large	60.0	23.8	25.4	50.4	40.1
	+ LRAT (Ours)	**63.3** (+5.5%)	**20.2**	**29.0** (+14.2%)	**56.1** (+11.3%)	**39.1**
Tongyi-DeepResearch (30B)	Qwen3-Emb	52.7	26.7	17.8	49.2	42.9
	+ LRAT (Ours)	**68.0** (+29.0%)	**20.7**	**23.7** (+33.1%)	**60.7** (+23.4%)	**41.0**
	E5-Large	56.7	25.1	20.7	54.8	42.4
	+ LRAT (Ours)	**68.0** (+19.9%)	**21.5**	**23.9** (+15.5%)	**61.8** (+12.8%)	**41.4**
II. Generalist Agentic Foundation Models						
GPT-OSS (120B)	Qwen3-Emb	40.0	34.9	9.0	43.7	45.4
	+ LRAT (Ours)	**47.0** (+17.5%)	**30.5**	**12.1** (+34.4%)	**56.4** (+29.1%)	**45.2**
	E5-Large	41.7	33.9	10.8	50.1	44.8
	+ LRAT (Ours)	**50.7** (+21.6%)	**29.7**	**13.1** (+21.3%)	**56.0** (+11.8%)	**44.6**
MiniMax-M2.1 (229B)	Qwen3-Emb	58.7	21.4	38.2	57.2	30.8
	+ LRAT (Ours)	**78.3** (+33.4%)	**14.7**	**48.3** (+26.4%)	**69.2** (+21.0%)	**28.3**
	E5-Large	64.0	18.9	46.4	64.9	29.1
	+ LRAT (Ours)	**75.0** (+17.2%)	**14.8**	**48.7** (+5.0%)	**69.7** (+7.4%)	**28.9**
GLM-4.7 (358B)	Qwen3-Emb	67.7	27.5	43.9	66.6	45.5
	+ LRAT (Ours)	**82.0** (+21.1%)	**18.5**	**54.6** (+24.4%)	**77.8** (+16.8%)	**44.6**
	E5-Large	73.7	24.2	46.4	68.7	**44.6**
	+ LRAT (Ours)	**81.7** (+10.9%)	**19.5**	**50.6** (+9.1%)	**76.3** (+11.1%)	44.8

6. Experiments

In this section, we conduct extensive experiments to evaluate the effectiveness of our proposed LRAT framework in agentic search settings.

6.1. Experimental Setup

6.1.1. Benchmarks

We evaluate LRAT on two benchmarks to assess both in-domain and out-of-domain generalization. For **in-domain evaluation**, we use **InfoSeek-Eval** Luo et al. (2025), which consists of 300 multi-hop information-seeking queries that are strictly disjoint from all training data in Section 4.1.1. For **out-of-domain evaluation**, we additionally evaluate on **BrowseComp-Plus** Chen et al. (2025), a reproducible benchmark specifically designed for deep research agents. It contains 830 complex, human-authored questions that require multi-step reasoning and evidence aggregation. Following the official protocol, retrieval is performed over a corpus of 100,195 documents.

6.1.2. Backbones

For the retrieval backbone, we consider two representative and widely adopted dense retrievers with complementary architectures: **Multilingual-E5-Large-Instruct** Wang et al. (2024), an encoder-based model, and **Qwen3-Embedding-0.6B** Zhang et al. (2025), a decoder-based embedding model.

To evaluate downstream impact in realistic agentic settings, we integrate these retrievers into multiple open-source search agents to ensure reproducibility. These include **task-optimized search agents** (**AgentCPM-Explore**-4B Chen et al. (2026), **WebExplore**-8B Liu et al. (2025), **Tongyi-DeepResearch**-30B Team et al. (2025b)) and **generalist agentic foundation models** (**GPT-OSS**-120B OpenAI (2025), **MiniMax-M2.1**-229B Team (2025), **GLM-4.7**-358B Team et al. (2025a)). This setup spans diverse retriever architectures and agent scales ranging from 4B to 358B parameters, enabling a comprehensive evaluation of LRAT across different agentic search systems.

6.1.3. Evaluation Metrics

Following the standard evaluation protocol of BrowseComp-Plus Chen et al. (2025), we adopt a multi-dimensional evaluation setup that assesses task outcome, execution efficiency, and retrieval quality. The primary metric is **Success Rate (SR)**, which is assessed by an automated LLM judge based on *Qwen3-30B-A3B-Thinking-2507* to verify answer correctness. To measure execution efficiency, we report the **Average Step Count**, where fewer steps indicate more direct information acquisition enabled by the retriever. To directly measure retrieval quality, we additionally report **Evidence Recall** on BrowseComp-Plus, defined as the proportion of tasks in which the annotated evidence document is successfully retrieved during the agent's execution. For InfoSeek-Eval, we report success rate and average step count only, as it does not provide the trace-level annotations required to compute evidence recall.

6.1.4. Implementation Details

We train the retriever with the FlagEmbedding framework FlagOpen Team (2023). The model is fine-tuned for 2 epochs, with a batch size of 32, a learning rate of 1e$-$6, and a maximum input length of 512 tokens. For InfoNCE loss, we use a group size of 10 and a temperature of 0.02. On the agent side, for reproducibility, we fix the random seed to 2025. Generation parameters are set to a temperature of 0.85, $\text{top}_p = 0.95$, and a presence penalty of 1.1. Retrieval configuration during execution is kept consistent with the training setup. Trajectories are limited to 50 turns per query during evaluation due to computational constraints.

6.2. Overall Performance

Table 2 reports the main results on both in-domain and out-of-domain benchmarks. Overall, LRAT yields consistent and substantial improvements across all retriever backbones and agent backbones, leading to higher task success, stronger evidence retrieval, and more efficient agent execution. Specifically, we have the following observations:

Improved Evidence Retrieval. On BrowseComp-Plus, LRAT significantly improves the retriever's ability to retrieve annotated evidence documents, as reflected by the recall metric. For both Qwen3-Embed and E5-Large, training with LRAT consistently increases evidence recall across all agents, with relative gains ranging from 7% to over 37%. This demonstrates that supervision derived from agent trajectories effectively enhances retrieval quality, enabling retrievers to better align with the information needs issued by agents.

End-to-end Gains in Task Success. Stronger retrieval quality directly translates into improved end-to-end performance. Across both in-domain and out-of-domain settings, agents equipped with LRAT-trained retrievers achieve substantially higher success rates than their baseline counterparts. These gains are consistent across task-optimized search agents and generalist agentic foundation models, indicating that LRAT generalizes well across agent architectures and parameter scales. Notably, improvements persist even for very large agents (e.g., 120B-358B), suggesting that retrieval

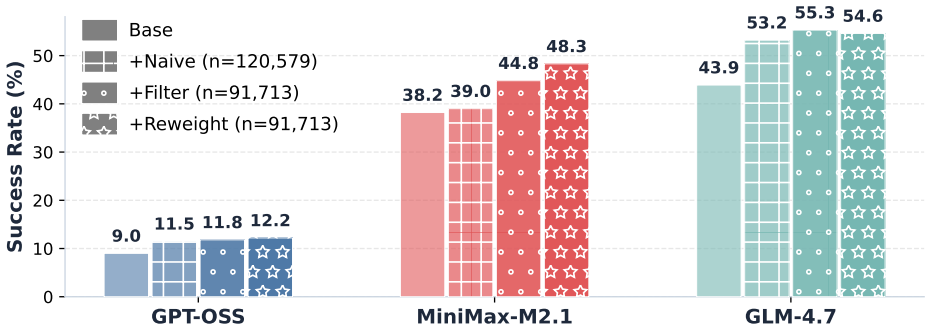


Figure 7 | Ablation study: components are incrementally added. Numbers n in parentheses show the amount of training data used for each variant of LRAT.

quality remains a critical bottleneck despite strong agent capabilities.

More Efficient Agent Execution. In addition to higher success rates, LRAT consistently reduces the average number of interaction steps required to solve a task. This effect is particularly pronounced on InfoSeek-Eval, where average step counts are reduced by up to $\sim 30\%$. The reduction in steps indicates that LRAT-trained retrievers provide more precise and useful evidence at each search step, allowing agents to satisfy their information needs with fewer exploratory interactions. As a result, agents achieve better performance while incurring fewer search and browse actions.

6.3. Ablation Study

We conduct an ablation study on BrowseComp-Plus using Qwen3-Embedding-0.6B as the retriever, with components of LRAT added incrementally. Unless otherwise specified, subsequent experiments use the same setting. The results in Figure 7 highlight the significance of each design:

"+Naive" refers to a variant that treats only the documents browsed by the agent as positive samples, while regarding all other documents as negatives. This strategy yields substantial performance gains, suggesting the absence of strong position bias in the agent's browsing process and confirming that unbrowsed documents can serve as reliable negative signals.

"+Filter" further introduces LLM-based filtering over the browsed documents, removing false positive documents that are accessed but do not meaningfully contribute to the agent's subsequent reasoning. The resulting improvement indicates that post-browse reasoning traces are important indicators of document usefulness, and leveraging them helps refine the quality of supervision signals.

"+Reweight" incorporates relevance intensity estimation by using the reasoning length of the agent as a proxy for importance. The resulting performance gains highlight the necessity of accounting for the heterogeneous contributions of different documents and empirically validate the effectiveness of LRAT in leveraging reasoning-aware supervision.

Overall, the ablation results empirically validate the progressive design of LRAT, showing that trajectory-derived supervision becomes increasingly effective as richer agent signals are mined and incorporated into retriever training.

6.4. Scalability and Robustness Analysis

To assess whether LRAT remains effective at realistic scales, we analyze its behavior under increasing training data and varying top-K retrieval budgets.

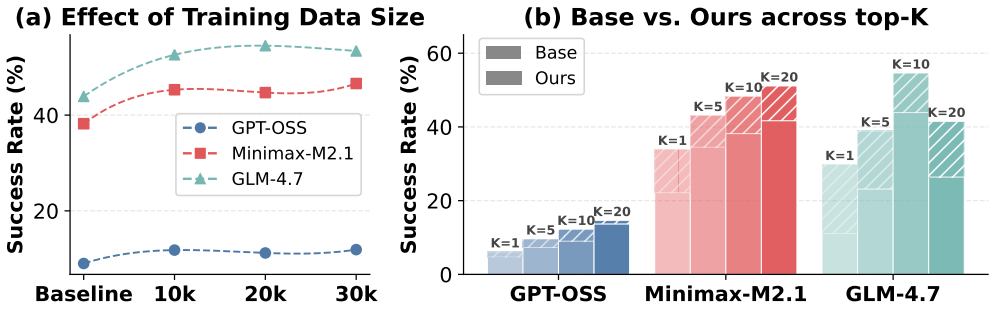


Figure 8 | Agent performance with varying training data sizes and retrieval top-K settings.

Table 3 | Trajectory correctness ablation. Both correct and incorrect trajectories use 10K examples each.

Training Data	GPT-OSS	MiniMax-M2.1	GLM-4.7
Base (*w/o* LRAT)	9.0	38.2	43.9
LRAT (*w/* Incorrect Trajectories)	10.7 (+18.9%)	43.6 (+14.1%)	50.6 (+15.3%)
LRAT (*w/* Correct Trajectories)	11.8 (+31.1%)	45.3 (+18.6%)	52.6 (+19.8%)

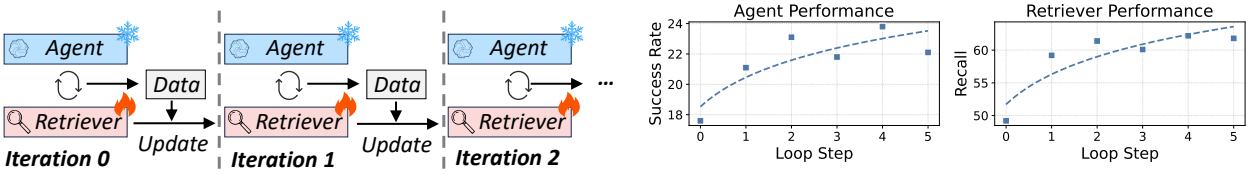
Training-time Scalability. We investigate whether retrievers trained with LRAT can consistently benefit from additional agent interaction data. Specifically, we collect 30K agent trajectories on InfoSeekQA using the Tongyi-DeepResearch agent with BM25, and train retrievers with progressively larger subsets of the collected data. As shown in Figure 8(a), agent success rates generally improve as the training dataset grows across all evaluated models, indicating that LRAT can effectively exploit larger volumes of agent trajectories and does not suffer from early performance saturation.

Inference-time Robustness. We further evaluate agent performance under different top-K retrieval settings by comparing the base retriever with the LRAT-enhanced retriever. As illustrated in Figure 8(b), increasing K does not always lead to monotonic performance improvements: while moderate retrieval budgets are beneficial, overly large K can degrade performance (e.g., GLM-4.7), likely due to increased noise and limited effective context capacity. Despite this, LRAT consistently outperforms the base retriever across all evaluated top-K values, demonstrating its robustness to varying retrieval budgets and its effectiveness under both information-scarce and noise-heavy retrieval conditions.

6.5. Data Flywheel Simulation and Analysis

Traditionally, human click logs are continuously exploited to iteratively improve retrievers, creating a self-sustaining data flywheel. We investigate whether a similar mechanism can arise under the LRAT framework in agentic search settings. Unlike benchmark evaluation, real user queries are open-ended, and agent trajectories are not always fully correct, raising the question of whether imperfect trajectories can still provide useful supervision. Our preliminary analysis suggests that this is indeed the case. As shown in Table 3, retrievers trained with both correct and incorrect trajectories consistently outperform the base retriever, although incorrect trajectories yield smaller gains. This indicates that even when an agent fails to produce the correct final answer, its intermediate interactions with the retriever still reflect meaningful judgments about document utility. Thus, we can include all collected trajectories when correctness labels are unavailable or unreliable.

The simulated data flywheel is illustrated in Figure 9a, where the retriever undergoes iterative updates through continuous agent interactions. At each step, the retriever collects trajectories from agent interactions and is updated before the next step, mimicking a realistic streaming



(a) Illustration of data flywheel simulation setting. (b) Performance of the data flywheel simulation.

Figure 9 | Data flywheel simulation in the single-column layout. Left: the iterative update setting used to collect and refresh retriever supervision. Right: the resulting performance trend, showing steady gains in both success rate and evidence recall across iterations.

environment. To ensure consistency with our main experiments while controlling computational cost, we adopt the Tongyi-DeepResearch agent and sample 10K queries from InfoSeekQA at each step. Evaluation is conducted using the same agent, reflecting a realistic deployment scenario with repeated retriever–agent interactions. The results in Figure 9b show steady improvements in both agent success rate and retriever recall across iterations, demonstrating that our method can reliably support iterative retriever updates and sustain a positive data flywheel. Moreover, performance is maintained or even improved in this streaming setting, highlighting the practical value of agent-based trajectory supervision for real-world retrieval systems.

7. Conclusion

This paper identifies a fundamental misalignment between human-centric retrieval training and the needs of agentic search, and formalizes *learning to retrieve from agent trajectories* as a new retrieval paradigm. Empirical analysis of deep research agent trajectories reveals that browsing behavior can serve as a reliable indicator of document utility, unbrowsed results can provide trustworthy negative signals, and post-browse reasoning traces can capture the intensity of document relevance during multi-step problem solving. A retrieval framework called LRAT is proposed to convert agent trajectories into supervision signals to train agent-aligned retrievers. Experiments on both in-domain and out-of-domain deep research benchmarks demonstrate the effectiveness of LRAT across diverse agent and retriever backbones. Empirical analysis also demonstrates that agent trajectories can support iterative retriever improvement, indicating the potential for a sustainable data flywheel driven by agent interactions. These findings highlight agent trajectories as a practical and scalable supervision source, and suggest a promising direction for advancing retrieval systems in the era of agentic search.

References

E. Agichtein, E. Brill, and S. Dumais. Improving web search ranking by incorporating user behavior information. In *Proceedings of the 29th annual international ACM SIGIR conference on Research and development in information retrieval*, pages 19–26, 2006a.

E. Agichtein, E. Brill, S. Dumais, and R. Ragno. Learning user interaction models for predicting web search result preferences. In *Proceedings of the 29th annual international ACM SIGIR conference on Research and development in information retrieval*, pages 3–10, 2006b.

R. Baeza-Yates, B. Ribeiro-Neto, et al. *Modern information retrieval*, volume 463. ACM press New York, 1999.

C. Burges, T. Shaked, E. Renshaw, A. Lazier, M. Deeds, N. Hamilton, and G. Hullender. Learning to

rank using gradient descent. In *Proceedings of the 22nd international conference on Machine learning*, pages 89–96, 2005.

C. J. Burges. From ranknet to lambdarank to lambdamart: An overview. *Learning*, 11(23-581):81, 2010.

Y. Cao, J. Xu, T.-Y. Liu, H. Li, Y. Huang, and H.-W. Hon. Adapting ranking svm to document retrieval. In *Proceedings of the 29th annual international ACM SIGIR conference on Research and development in information retrieval*, pages 186–193, 2006.

Z. Cao, T. Qin, T.-Y. Liu, M.-F. Tsai, and H. Li. Learning to rank: from pairwise approach to listwise approach. In *Proceedings of the 24th international conference on Machine learning*, pages 129–136, 2007.

H. Chen, X. Cong, S. Fan, Y. Fu, Z. Gong, Y. Lu, Y. Li, B. Niu, C. Pan, Z. Song, H. Wang, Y. Wu, Y. Wu, Z. Xie, Y. Yan, Z. Zhang, Y. Lin, Z. Liu, and M. Sun. Agentcpm-explore: An end-to-end infrastructure for training and evaluating llm agents, 2026. URL https://github.com/OpenBMB/AgentCPM.

Z. Chen, X. Ma, S. Zhuang, P. Nie, K. Zou, A. Liu, J. Green, K. Patel, R. Meng, M. Su, S. Sharifymoghaddam, Y. Li, H. Hong, X. Shi, X. Liu, N. Thakur, C. Zhang, L. Gao, W. Chen, and J. Lin. Browsecomp-plus: A more fair and transparent evaluation benchmark of deep-research agent. *arXiv preprint arXiv:2508.06600*, 2025.

G. G. Chowdhury. *Introduction to modern information retrieval*. Facet publishing, 2010.

N. Craswell, O. Zoeter, M. Taylor, and B. Ramsey. An experimental comparison of click position-bias models. In *Proceedings of the 2008 international conference on web search and data mining*, pages 87–94, 2008.

W. B. Croft, D. Metzler, and T. Strohman. *Search engines: Information retrieval in practice*, volume 520. Addison-Wesley Reading, 2010.

S. Dai, W. Wang, L. Pang, J. Xu, S.-K. Ng, J.-R. Wen, and T.-S. Chua. Next-search: Rebuilding user feedback ecosystem for generative ai search. In *Proceedings of the 48th International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 3922–3931, 2025.

G. Dong, Y. Zhu, C. Zhang, Z. Wang, J.-R. Wen, and Z. Dou. Understand what llm needs: Dual preference alignment for retrieval-augmented generation. In *Proceedings of the ACM on Web Conference 2025*, pages 4206–4225, 2025.

FlagOpen Team. Flagembedding: A powerful toolkit for retrieval and retrieval-augmented llms. https://github.com/FlagOpen/FlagEmbedding, 2023.

L. A. Granka, T. Joachims, and G. Gay. Eye-tracking analysis of user behavior in www search. In *Proceedings of the 27th annual international ACM SIGIR conference on Research and development in information retrieval*, pages 478–479, 2004.

M. Gutmann and A. Hyvärinen. Noise-contrastive estimation: A new estimation principle for unnormalized statistical models. In *Proceedings of the thirteenth international conference on artificial intelligence and statistics*, pages 297–304. JMLR Workshop and Conference Proceedings, 2010.

Y. Huang, Y. Chen, H. Zhang, K. Li, M. Fang, L. Yang, X. Li, L. Shang, S. Xu, J. Hao, et al. Deep research agents: A systematic examination and roadmap. *arXiv preprint arXiv:2506.18096*, 2025.

B. Jin, H. Zeng, Z. Yue, J. Yoon, S. O. Arik, D. Wang, H. Zamani, and J. Han. Search-r1: Training LLMs to reason and leverage search engines with reinforcement learning. In *Second Conference on Language Modeling*, 2025. URL https://openreview.net/forum?id=Rwhi91ideu.

T. Joachims. Optimizing search engines using clickthrough data. In *Proceedings of the eighth ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 133–142, 2002.

T. Joachims, L. Granka, B. Pan, H. Hembrooke, and G. Gay. Accurately interpreting clickthrough data as implicit feedback. In *Proceedings of the 28th annual international ACM SIGIR conference on Research and development in information retrieval*, page 154–161, 2005.

Z. Ke, W. Kong, C. Li, M. Zhang, Q. Mei, and M. Bendersky. Bridging the preference gap between retrievers and LLMs. In L.-W. Ku, A. Martins, and V. Srikumar, editors, *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, Aug. 2024.

D. Kelly and N. J. Belkin. Display time as implicit feedback: understanding task effects. In *Proceedings of the 27th annual international ACM SIGIR conference on Research and development in information retrieval*, pages 377–384, 2004.

Y. Kim, A. Hassan, R. W. White, and I. Zitouni. Modeling dwell time to predict click-level satisfaction. In *Proceedings of the 7th ACM international conference on Web search and data mining*, pages 193–202, 2014.

P. Li, Q. Wu, and C. Burges. Mcrank: Learning to rank using multiple classification and gradient boosting. *Advances in neural information processing systems*, 20, 2007.

X. Li, G. Dong, J. Jin, Y. Zhang, Y. Zhou, Y. Zhu, P. Zhang, and Z. Dou. Search-o1: Agentic search-enhanced large reasoning models. In C. Christodoulopoulos, T. Chakraborty, C. Rose, and V. Peng, editors, *Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing*, pages 5420–5438, Suzhou, China, Nov. 2025. Association for Computational Linguistics. ISBN 979-8-89176-332-6. doi: 10.18653/v1/2025.emnlp-main.276. URL https://aclanthology.org/2025.emnlp-main.276/.

J. Liu, Y. Li, C. Zhang, J. Li, A. Chen, K. Ji, W. Cheng, Z. Wu, C. Du, Q. Xu, J. Song, Z. Zhu, W. Chen, P. Zhao, and J. He. Webexplorer: Explore and evolve for training long-horizon web agents, 2025. URL https://arxiv.org/abs/2509.06501.

T.-Y. Liu et al. Learning to rank for information retrieval. *Foundations and Trends® in Information Retrieval*, 3(3):225–331, 2009.

Y. Liu, X. Xie, C. Wang, J.-Y. Nie, M. Zhang, and S. Ma. Time-aware click model. *ACM Transactions on Information Systems (TOIS)*, 35(3):1–24, 2016.

K. Luo, H. Qian, Z. Liu, Z. Xia, S. Xiao, S. Bao, J. Zhao, and K. Liu. Infoflow: Reinforcing search agent via reward density optimization. *arXiv preprint arXiv:2510.26575*, 2025.

C. D. Manning. *An introduction to information retrieval*. 2009.

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

C. Qu, S. Dai, H. Cai, Y. Cheng, J. Xu, S. Wang, and D. Yin. Uplift-RAG: Uplift-driven knowledge preference alignment for retrieval-augmented generation. In C. Christodoulopoulos, T. Chakraborty, C. Rose, and V. Peng, editors, *Findings of the Association for Computational Linguistics: EMNLP 2025*, pages 9632–9644, Suzhou, China, Nov. 2025. Association for Computational Linguistics. ISBN 979-8-89176-335-7. doi: 10.18653/v1/2025.findings-emnlp.511. URL https://aclanthology.org/2025.findings-emnlp.511/.

F. Radlinski, M. Kurup, and T. Joachims. How does clickthrough data reflect retrieval quality? In *Proceedings of the 17th ACM conference on Information and knowledge management*, pages 43–52, 2008.

S. Robertson, H. Zaragoza, et al. The probabilistic relevance framework: Bm25 and beyond. *Foundations and Trends® in Information Retrieval*, 3(4):333–389, 2009.

A. Salemi and H. Zamani. Evaluating retrieval quality in retrieval-augmented generation. In *Proceedings of the 47th International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 2395–2400, 2024.

X. Shen, B. Tan, and C. Zhai. Implicit user modeling for personalized search. In *Proceedings of the 14th ACM international conference on Information and knowledge management*, pages 824–831, 2005.

W. Shi, S. Min, M. Yasunaga, M. Seo, R. James, M. Lewis, L. Zettlemoyer, and W.-t. Yih. REPLUG: Retrieval-augmented black-box language models. In *Proceedings of the 2024 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 1: Long Papers)*, 2024.

Z. Shi, Y. Chen, H. Li, W. Sun, S. Ni, Y. Lyu, R.-Z. Fan, B. Jin, Y. Weng, M. Zhu, et al. Deep research: A systematic survey. *arXiv preprint arXiv:2512.02038*, 2025.

H. Song, J. Jiang, Y. Min, J. Chen, Z. Chen, W. X. Zhao, L. Fang, and J.-R. Wen. R1-searcher: Incentivizing the search capability in llms via reinforcement learning. *arXiv preprint arXiv:2503.05592*, 2025.

G. Team, A. Zeng, X. Lv, Q. Zheng, Z. Hou, B. Chen, C. Xie, C. Wang, D. Yin, H. Zeng, J. Zhang, K. Wang, L. Zhong, M. Liu, R. Lu, S. Cao, X. Zhang, X. Huang, Y. Wei, Y. Cheng, Y. An, Y. Niu, Y. Wen, Y. Bai, Z. Du, Z. Wang, Z. Zhu, B. Zhang, B. Wen, B. Wu, B. Xu, C. Huang, C. Zhao, C. Cai, C. Yu, C. Li, C. Ge, C. Huang, C. Zhang, C. Xu, C. Zhu, C. Li, C. Yin, D. Lin, D. Yang, D. Jiang, D. Ai, E. Zhu, F. Wang, G. Pan, G. Wang, H. Sun, H. Li, H. Li, H. Hu, H. Zhang, H. Peng, H. Tai, H. Zhang, H. Wang, H. Yang, H. Liu, H. Zhao, H. Liu, H. Yan, H. Liu, H. Chen, J. Li, J. Zhao, J. Ren, J. Jiao, J. Zhao, J. Yan, J. Wang, J. Gui, J. Zhao, J. Liu, J. Li, J. Li, J. Lu, J. Wang, J. Yuan, J. Li, J. Du, J. Du, J. Liu, J. Zhi, J. Gao, K. Wang, L. Yang, L. Xu, L. Fan, L. Wu, L. Ding, L. Wang, M. Zhang, M. Li, M. Xu, M. Zhao, M. Zhai, P. Du, Q. Dong, S. Lei, S. Tu, S. Yang, S. Lu, S. Li, S. Li, Shuang-Li, S. Yang, S. Yi, T. Yu, W. Tian, W. Wang, W. Yu, W. L. Tam, W. Liang, W. Liu, X. Wang, X. Jia, X. Gu, X. Ling, X. Wang, X. Fan, X. Pan, X. Zhang, X. Zhang, X. Fu, X. Zhang, Y. Xu, Y. Wu, Y. Lu, Y. Wang, Y. Zhou, Y. Pan, Y. Zhang, Y. Wang, Y. Li, Y. Su, Y. Geng, Y. Zhu, Y. Yang, Y. Li, Y. Wu, Y. Li, Y. Liu, Y. Wang, Y. Li, Y. Zhang, Z. Liu, Z. Yang, Z. Zhou, Z. Qiao, Z. Feng, Z. Liu, Z. Zhang, Z. Wang, Z. Yao, Z. Wang, Z. Liu, Z. Chai, Z. Li, Z. Zhao, W. Chen, J. Zhai, B. Xu, M. Huang, H. Wang, J. Li, Y. Dong, and J. Tang. Glm-4.5: Agentic, reasoning, and coding (arc) foundation models, 2025a. URL https://arxiv.org/abs/2508.06471.

M. Team. Minimax-m2.1: Significantly enhanced multi-language programming, built for real-world complex tasks. https://huggingface.co/MiniMaxAI/MiniMax-M2.1, 2025.

T. D. Team, B. Li, B. Zhang, D. Zhang, F. Huang, G. Li, G. Chen, H. Yin, J. Wu, J. Zhou, et al. Tongyi deepresearch technical report. *arXiv preprint arXiv:2510.24701*, 2025b.

G. Wang, S. Dai, G. Ye, Z. Gan, W. Yao, Y. Deng, X. Wu, and Z. Ying. Information gain-based policy optimization: A simple and effective approach for multi-turn llm agents. *arXiv preprint arXiv:2510.14967*, 2025.

L. Wang, N. Yang, X. Huang, L. Yang, R. Majumder, and F. Wei. Multilingual e5 text embeddings: A technical report. *arXiv preprint arXiv:2402.05672*, 2024.

R. W. White. Advancing the search frontier with ai agents. *Communications of the ACM*, 67(9):54–65, 2024.

Y. Xi, J. Lin, Y. Xiao, Z. Zhou, R. Shan, T. Gao, J. Zhu, W. Liu, Y. Yu, and W. Zhang. A survey of llm-based deep search agents: Paradigm, optimization, evaluation, and challenges. *arXiv preprint arXiv:2508.05668*, 2025.

Z. Xia, K. Luo, H. Qian, and Z. Liu. Open data synthesis for deep research. *arXiv preprint arXiv:2509.00375*, 2025.

J. Xu and H. Li. Adarank: a boosting algorithm for information retrieval. In *Proceedings of the 30th annual international ACM SIGIR conference on Research and development in information retrieval*, pages 391–398, 2007.

R. Xu and J. Peng. A comprehensive survey of deep research: Systems, methodologies, and applications. *arXiv preprint arXiv:2506.12594*, 2025.

A. Yang, A. Li, B. Yang, B. Zhang, B. Hui, B. Zheng, B. Yu, C. Gao, C. Huang, C. Lv, et al. Qwen3 technical report. *arXiv preprint arXiv:2505.09388*, 2025.

S. Yao, J. Zhao, D. Yu, N. Du, I. Shafran, K. R. Narasimhan, and Y. Cao. React: Synergizing reasoning and acting in language models. In *The eleventh international conference on learning representations*, 2022.

H. Zamani and M. Bendersky. Stochastic rag: End-to-end retrieval-augmented generation through expected utility maximization. In *Proceedings of the 47th International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 2641–2646, 2024.

H. Zhang, R. Zhang, J. Guo, M. de Rijke, Y. Fan, and X. Cheng. Are large language models good at utility judgments? In *Proceedings of the 47th International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 1941–1951, 2024.

Y. Zhang, M. Li, D. Long, X. Zhang, H. Lin, B. Yang, P. Xie, A. Yang, D. Liu, J. Lin, F. Huang, and J. Zhou. Qwen3 embedding: Advancing text embedding and reranking through foundation models. *arXiv preprint arXiv:2506.05176*, 2025.

Y. Zheng, D. Fu, X. Hu, X. Cai, L. Ye, P. Lu, and P. Liu. Deepresearcher: Scaling deep research via reinforcement learning in real-world environments, 2025. URL https://arxiv.org/abs/2504.03160.