SECTION 4

Utility in Agentic RAG

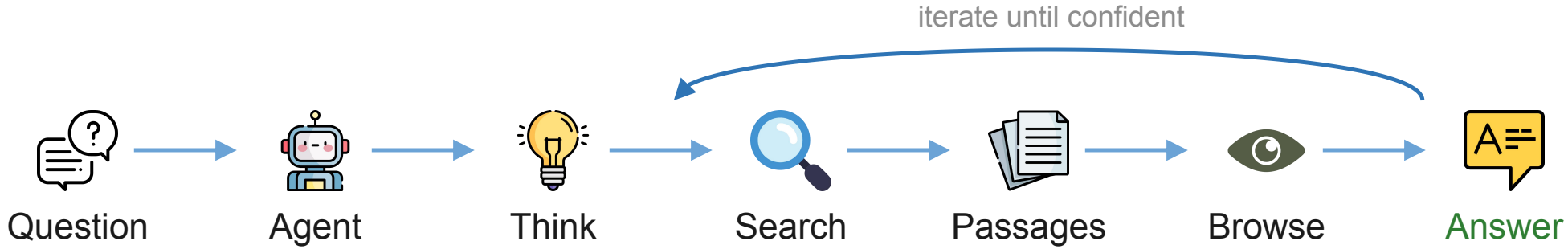
When utility becomes state-dependent — query, document, and action utility across an agent's trajectory.

4

Static RAG vs. Agentic RAG

STATIC RAG — retrieve once, then generate an answer

AGENTIC RAG — reason, search, and verify iteratively across steps

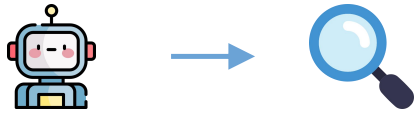


Utility becomes **dynamic** —
the information need changes over steps, so what counts as useful changes too.

Three Utilities in Agentic RAG

① Query-side Utility

What query should it issue for the current state?

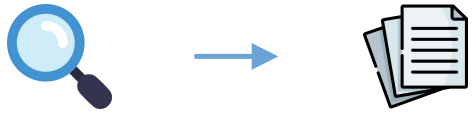


[query: ?]

a state-relevant query

② Document-side Utility

Does the retrieved evidence **actually help**?



useful? √ / ✗

local now & global answer

③ Action-side Utility

Which **action next** — incl. when to retrieve, or stop?

[Search / Browse / Find / Bash / Answer …]

A fourth, **trajectory-level utility/reward** binds the three — it supplies their training signal.

The Three Utilities — Map

LLM Information Need

The query-side question —
what query should the model issue for its current state?

The Three Utilities — Map

LLM Information Need

The query-side question — what query should the model issue for its current state?

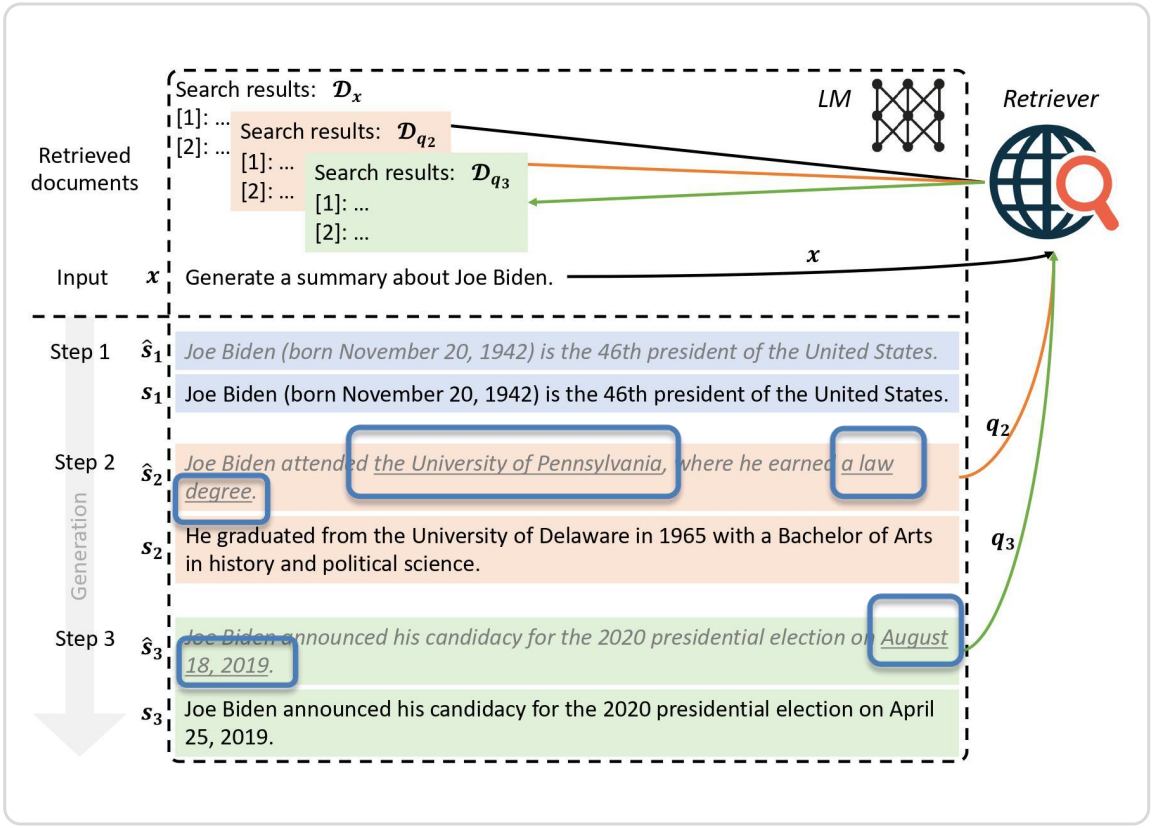
> Its companion — **when to retrieve** (retrieve vs. keep reasoning) — is an **action-side** decision; we give it its own treatment in later. They stay coupled — the same internal-state signal (FLARE, DRAGIN, Search-R1) informs both.

FLARE

QUERY-SIDE — WHERE KNOWLEDGE IS MISSING

> LLMs are fairly **well-calibrated**: low token probability is a proxy for missing knowledge.

Low-probability tokens in the next generated sentence pinpoint **where the model's knowledge is missing** — the gap a retrieval should go fill.



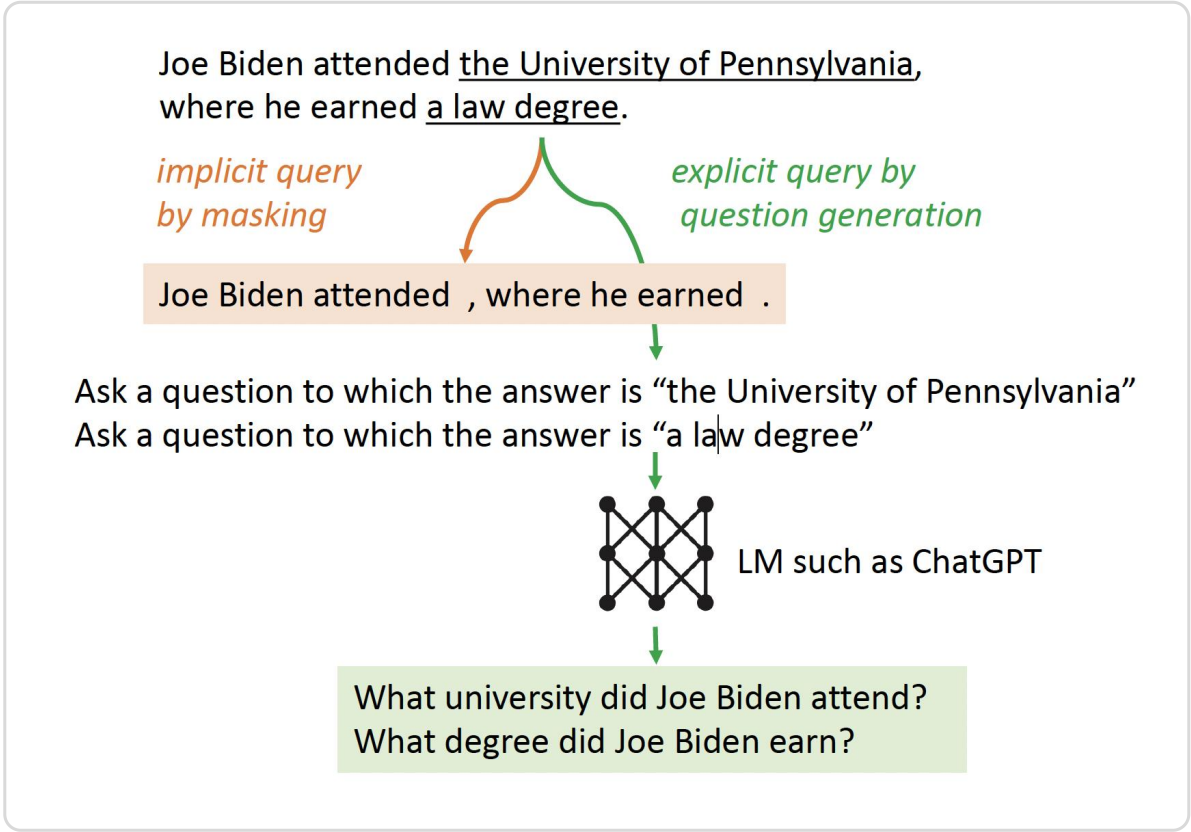
Jiang et al. Active Retrieval Augmented Generation. EMNLP, 2023.

FLARE — What Query to Issue

WHAT QUERY?

- **Implicit query** — mask the low-confidence tokens in the predicted next sentence.
- **Explicit query** — prompt the LLM to ask a question that targets the low-confidence span.

> Query utility = **filling the model's uncertainty** at this generation step.



Jiang et al. Active Retrieval Augmented Generation. EMNLP, 2023.

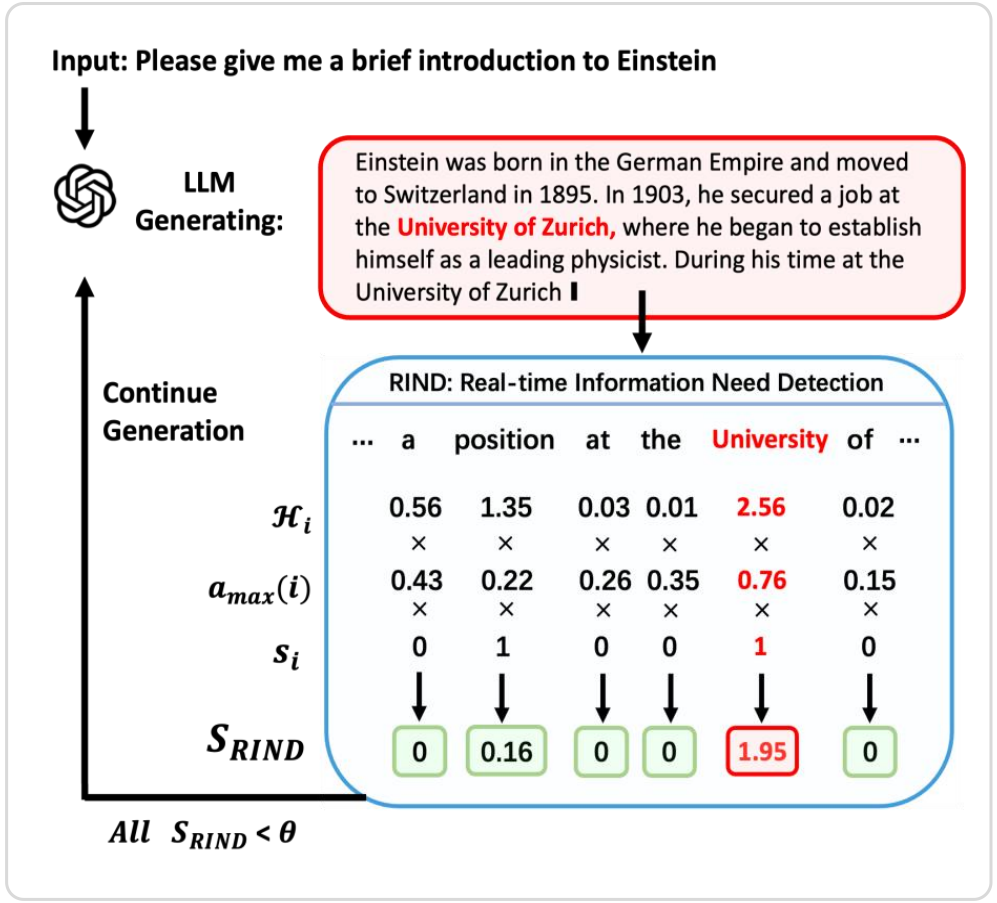
DRAGIN

QUERY-SIDE — FILTERING THE REAL NEED

> Not every uncertain token is a real need — stop words and pronouns are uncertain but meaningless.

What worth querying?

- High entropy
- High attention
- Significant (not a stopword/pronoun)



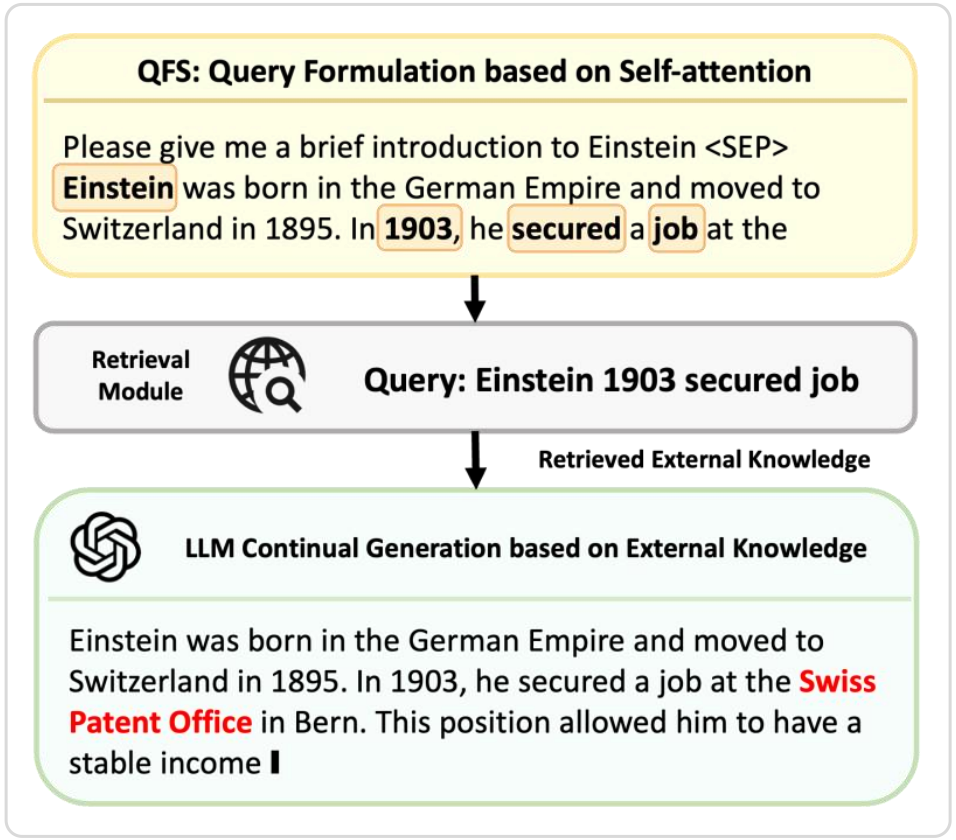
Su et al. DRAGIN: Dynamic Retrieval Augmented Generation based on the Real-time Information Needs of LLMs. ACL, 2024.

DRAGIN — What Query to Issue

WHAT QUERY?

Take the last layer‘s **attention** scores, keep the **top-n** most-attended tokens as the query

> The query reflects **what the model is actually attending to** — not the whole generated sentence.



Su et al. DRAGIN: Dynamic Retrieval Augmented Generation based on the Real-time Information Needs of LLMs. ACL, 2024.

Search-R1 — Learned Search Actions

The LLM learns to **write its own query** — no hand-designed rule.

What (query-side)

- it composes the query in <search>.

When (action-side)

- it also emits <search> itself.

THE ROLLOUT FORMAT

<think> reasoning </think>
<search> query **</search>**
<information> docs **</information>**
<answer> final answer **</answer>**

Jin et al. Search-R1: Training LLMs to Reason and Leverage Search Engines with Reinforcement Learning. COLM, 2025.

Search-R1 — Learned Search Actions

The LLM learns to **write its own query** — no hand-designed rule.

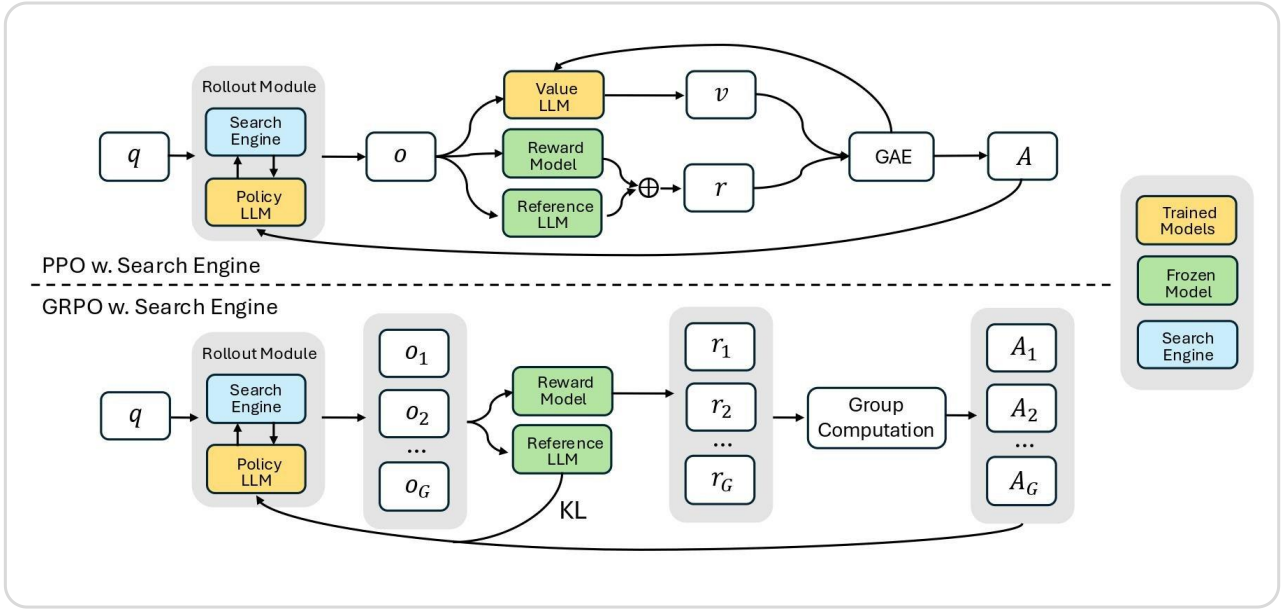
→ **learned through an end-to-end RL policy** (PPO / GRPO) (trajectory reward)

What (query-side)

- it composes the query in <search>.

When (action-side)

- it also emits <search> itself.



Jin et al. Search-R1: Training LLMs to Reason and Leverage Search Engines with Reinforcement Learning. COLM, 2025.

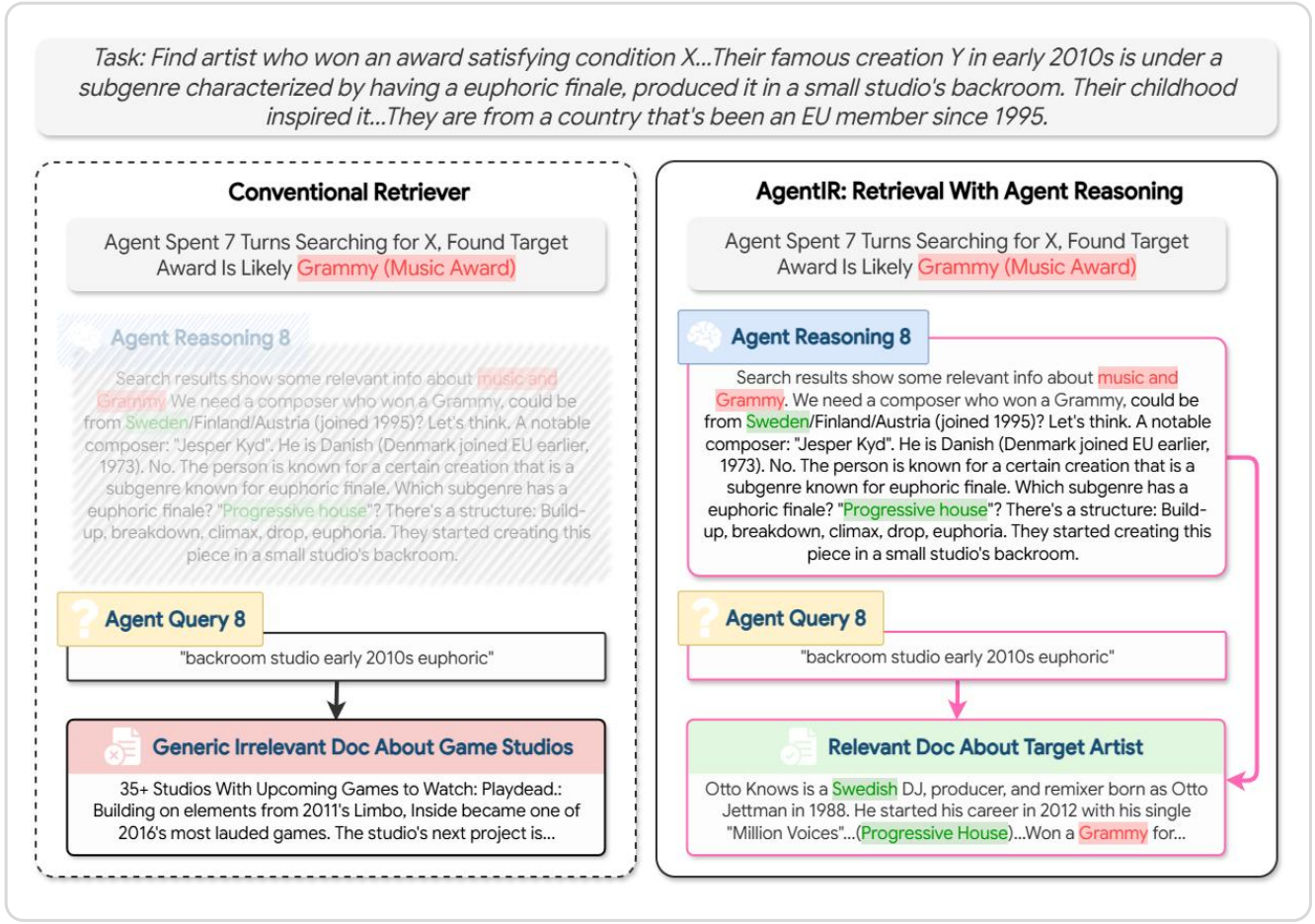
AgentIR

THE PROBLEM

An agent query is often **under-specified**. — the reasoning chain that produced it carries the **real search intent**

> Jointly embed the query and the agent's **reasoning trace alongside its query**

A type of query expansion with more specified context



Chen et al. AgentIR: Reasoning-Aware Retrieval for Deep Research Agents. arXiv, 2026.

Query-side Utility — Recap

A query is useful if it retrieves evidence that:

- fills current **knowledge gaps**
- supports the **next reasoning step**
- resolves **ambiguity or contradictions**
- **verifies** a claim
- improves the **final answer**

> **Query rewriting = utility optimization —** beyond lexical match, toward evidence acquisition.
> And the query to issue can be learned by RL over the trajectory.

The Three Utilities — Map

DOCUMENT-SIDE UTILITY

Query-side and document-side utility are coupled —
a query is useful because of the documents it brings back.

good queries → **relevant documents** → better reasoning

Document-side Utility

THE RETRIEVER OBJECTIVE SHIFTS

Document relevance ≠ document utility in agentic search.

Revisiting Text Ranking

A relevant document may pack 8K tokens — but only **~200** serve the current step.

> **Passage-level units** consistently beat document-level — **higher utility density**, more rounds, no information loss.

Retriever	Passage corpus				Document corpus			
	#Search	Recall	Acc.	Comp.	#Search	Recall	Acc.	Comp.
BM25	30.97	**0.616**	**0.572**	0.968	32.22	0.366	0.259	0.805
SPLADE-v3	32.75	0.545	0.516	0.980	28.95	<u>0.628</u>	<u>0.476</u>	0.824
RepLLaMA	36.13	0.449	0.406	0.961	30.69	0.514	0.363	0.786
Qwen3-Embed-8B	37.83	0.470	0.417	0.963	30.14	0.570	0.421	0.821
ColBERTv2	32.88	<u>0.552</u>	<u>0.521</u>	0.978	27.13	**0.633**	**0.481**	0.855

Meng et al. Revisiting Text Ranking in Deep Research. SIGIR, 2026.

Revisiting Text Ranking

A re-ranker lifts **recall** and **accuracy** while cutting the number of search calls.

> Re-ranking restores per-token **utility density** before the passages reach the agent.

Re-ranker	BM25				SPLADE-v3				Qwen3-Embed-8B			
	#Search	Recall	Acc.	Comp.	#Search	Recall	Acc.	Comp.	#Search	Recall	Acc.	Comp.
no re-ranking	30.97	0.616	0.572	0.980	32.75	0.545	0.516	0.980	37.83	0.470	0.417	0.963
monoT5 ($d = 10$)	29.89	0.660	0.631	0.971	31.58	0.630	0.599	0.965	35.64	0.524	0.471	0.969
monoT5 ($d = 20$)	28.96	0.701	0.674	0.960	30.07	0.647	0.598	0.974	34.16	0.570	0.541	0.959
monoT5 ($d = 50$)	27.57	**0.716**	**0.689**	0.974	29.72	0.689	0.646	0.971	32.94	0.614	0.559	0.952
RankLLaMA ($d = 10$)	29.85	0.657	0.617	0.980	31.30	0.632	0.595	0.964	36.07	0.508	0.465	0.964
RankLLaMA ($d = 20$)	29.03	0.681	0.655	0.971	30.82	0.646	0.605	0.981	34.25	0.553	0.494	0.961
RankLLaMA ($d = 50$)	27.10	0.710	0.678	0.977	28.96	0.691	**0.663**	0.959	32.85	0.613	**0.568**	0.964

A 20B model + a 3B re-ranker matches a full GPT-5 agent — at a fraction of the cost.

Meng et al. Revisiting Text Ranking in Deep Research. SIGIR, 2026.

Revisiting Text Ranking

Agent queries follow **web-search style**: keywords, quoted phrases, exact strings.

"90'+7" attendance 61700

"90'+4" football match attendance

BM25 outperforms neural retrievers:

Retriever	Passage corpus				Document corpus			
	#Search	Recall	Acc.	Comp.	#Search	Recall	Acc.	Comp.
BM25	30.97	**0.616**	**0.572**	0.968	32.22	0.366	0.259	0.805
SPLADE-v3	32.75	0.545	0.516	0.980	28.95	<u>0.628</u>	<u>0.476</u>	0.824
RepLLaMA	36.13	0.449	0.406	0.961	30.69	0.514	0.363	0.786
Qwen3-Embed-8B	37.83	0.470	0.417	0.963	30.14	0.570	0.421	0.821
ColBERTv2	32.88	<u>0.552</u>	<u>0.521</u>	0.978	27.13	**0.633**	**0.481**	0.855

Meng et al. Revisiting Text Ranking in Deep Research. SIGIR, 2026.

Revisiting Text Ranking

FIXING THE MISMATCH FOR DENSE

Keyword queries starve dense encoders. **Q2Q** rewrites them into the **questions** they trained on.

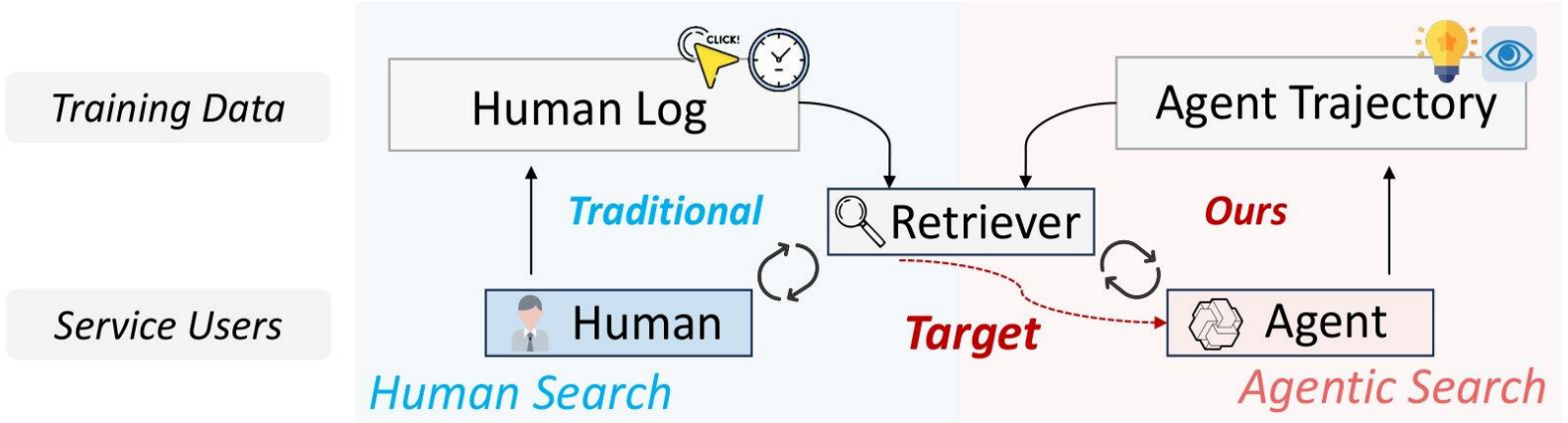
raw: "61,880" football attendance
Q2Q: What match drew an attendance of 61,880?

Retriever	Query	#Search	Recall	Acc.	Comp.
BM25	Raw	30.97	**0.616**	0.572	0.968
	Q2Q (Q)	31.88	0.593	**0.578**	0.977
	Q2Q (Q+R)	32.15	0.583	0.557	0.974
SPLADE-v3	Raw	32.75	0.545	0.516	0.980
	Q2Q (Q)	32.88	0.550	0.510	0.976
	Q2Q (Q+R)	31.70	**0.585***	**0.557***	0.983
Qwen3 -Embed-8B	Raw	37.83	0.470	0.417	0.963
	Q2Q (Q)	37.42	0.457	0.404	0.969
	Q2Q (Q+R)	35.82	**0.507***	**0.459***	0.965

Meng et al. Revisiting Text Ranking in Deep Research. SIGIR, 2026.

LRAT

Human search logs give retrievers one cheap signal — a **click = a positive**. Agents don't click; they issue a **[Browse]** action — LRAT reads it as the agent's click.



Browsed → positive

the agent opened it — a naive positive.

Unbrowsed → negative

left unopened in the same set — a naive negative.

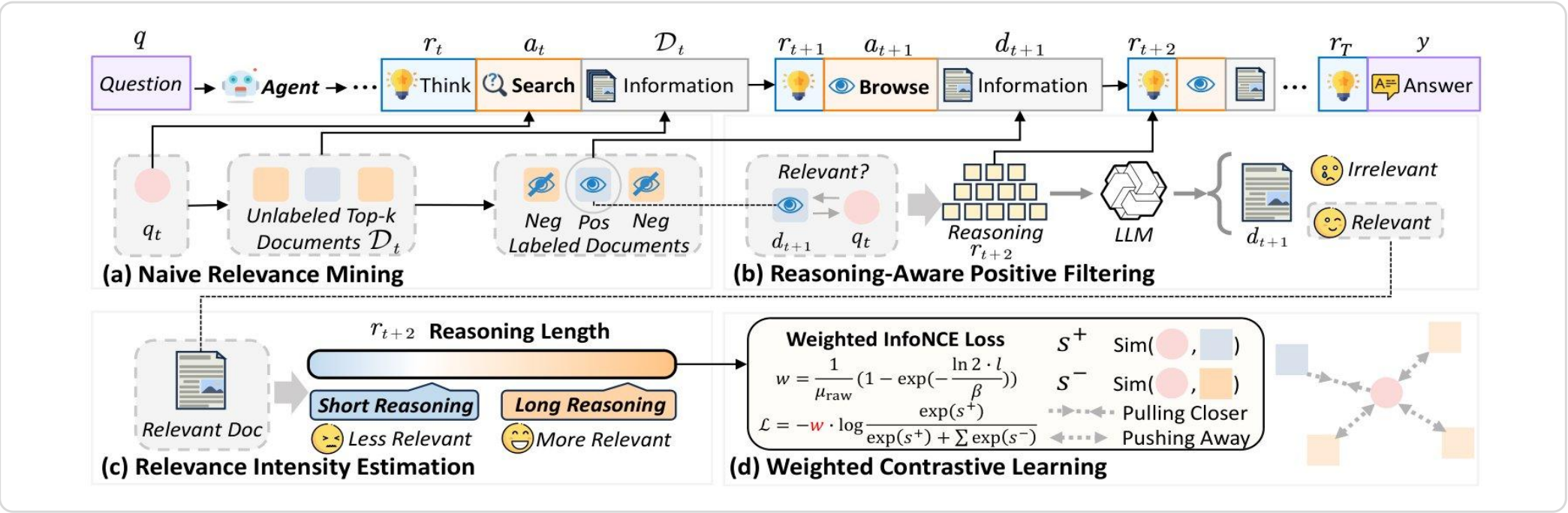
Zhou et al. Learning to Retrieve from Agent Trajectories. SIGIR, 2026.

LRAT

Naive browse labels are noisy — **browsed ≠ actually useful**.

LRAT refines them two ways:

① **Verifier** — the reasoning must **use** the doc. ② **Intensity** — **longer reasoning ⇒ more useful**.



Zhou et al. Learning to Retrieve from Agent Trajectories. SIGIR, 2026.

LRAT

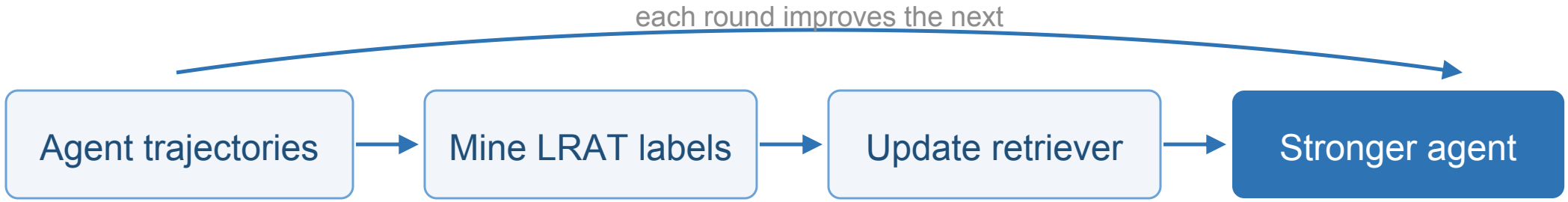
Agent Backbone	Retriever	InfoSeek-Eval (ID)		BrowseComp-Plus (OOD)		
		SR (↑)	Avg. Steps (↓)	SR (↑)	Recall (↑)	Avg. Steps (↓)
I. Task-Optimized Search Agents						
AgentCPM-Explore (4B)	Qwen3-Emb	40.3	38.0	13.5	23.2	40.7
	+ LRAT (Ours)	**55.7** (+38.2%)	**34.4**	**15.8** (+17.0%)	**32.0** (+37.9%)	**40.4**
	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**

Higher **success rate** & recall, fewer search steps — ID & OOD.

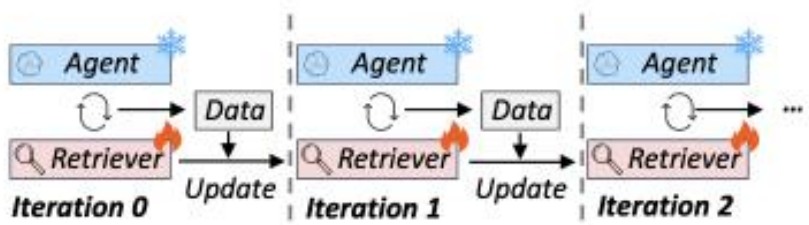
Zhou et al. Learning to Retrieve from Agent Trajectories. SIGIR, 2026.

LRAT — The Data Flywheel

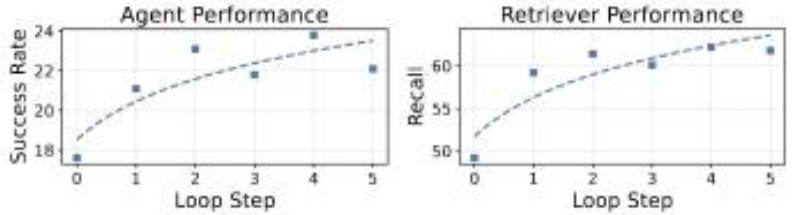
Labels → training → a stronger agent → better labels — a **positive data flywheel**.



FLYWHEEL SIMULATION



(a) Illustration of data flywheel simulation setting.

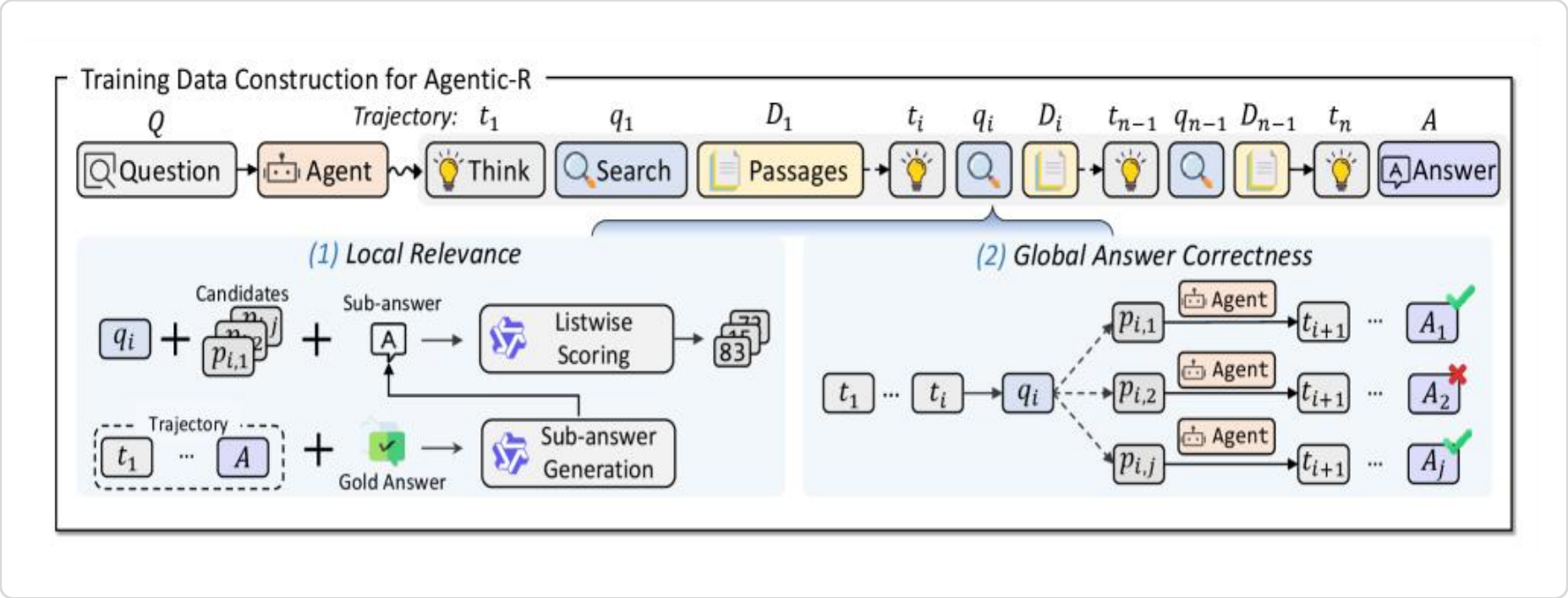


(b) Performance of the data flywheel simulation.

Zhou et al. Learning to Retrieve from Agent Trajectories. SIGIR, 2026.

Agentic-R

Intermediate queries have **no gold answers** — single-turn utility signals don't apply, and relevance could be insufficient to serve the agent reasoning towards the final answer. -> utility label for passages



Liu et al. Agentic-R: Learning to Retrieve for Agentic Search. ACL Findings, 2026.

Agentic-R — Dual-Signal Utility

With no gold answer per turn, one signal isn't enough — Agentic-R scores on **two**.

Local Relevance

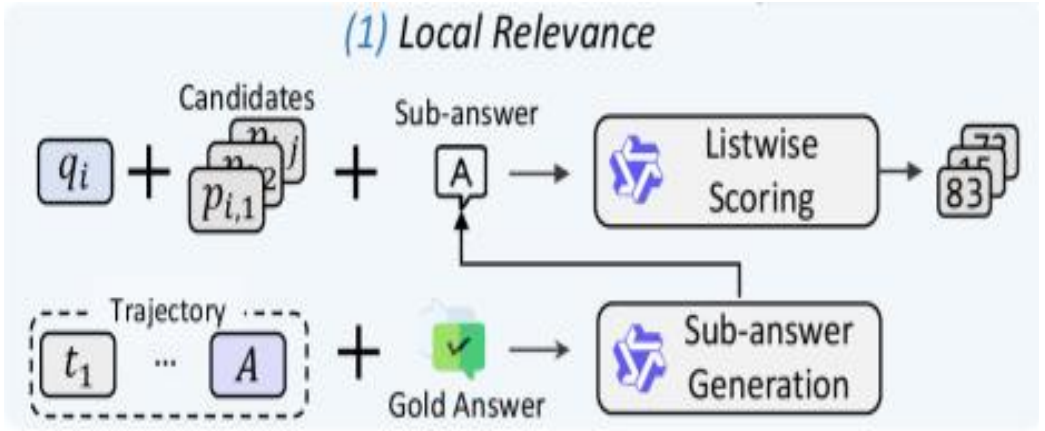
Does the passage address the **current sub-query**?

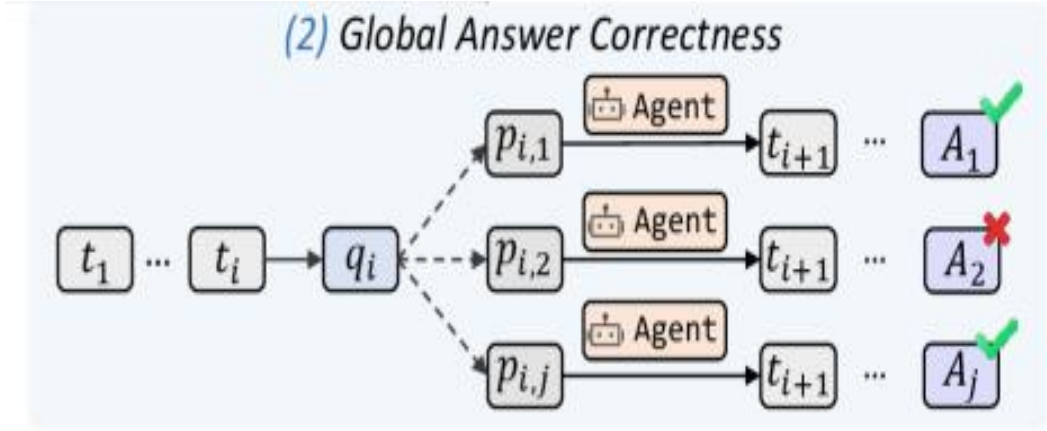
scored **0–100** by a strong LLM judge

Global Answer Correctness

Roll the agent forward on this passage — does it reach the **gold**

checked against the ground truth **(0 / 1)**





Liu et al. Agentic-R: Learning to Retrieve for Agentic Search. ACL Findings, 2026.

Agentic-R — Iterative Optimization

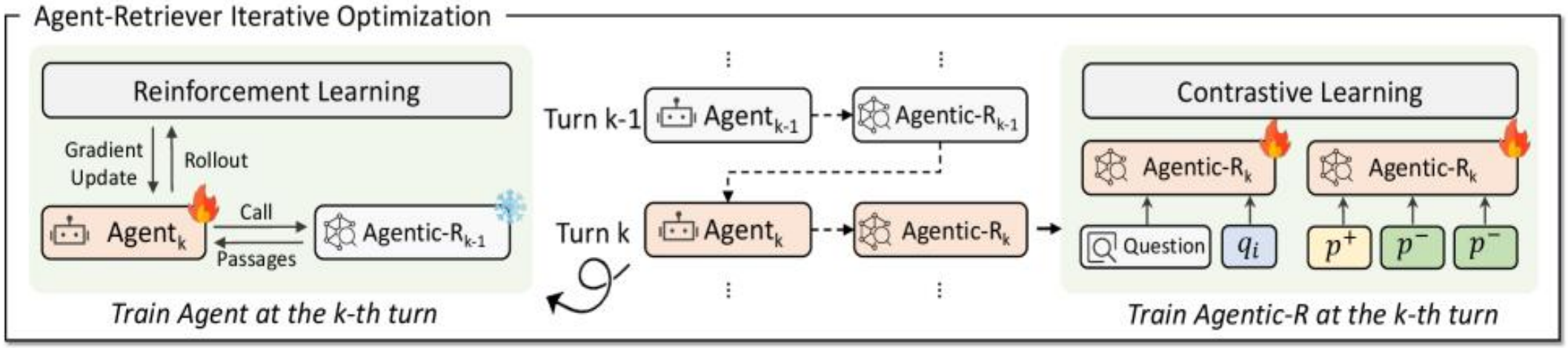
Agent and retriever train in an **alternating loop** — each freezes while the other learns.

1 Train Agent · retriever frozen

PPO with **EM on the final answer** as reward; the agent interacts to gather passages.

2 Train Retriever · agent frozen

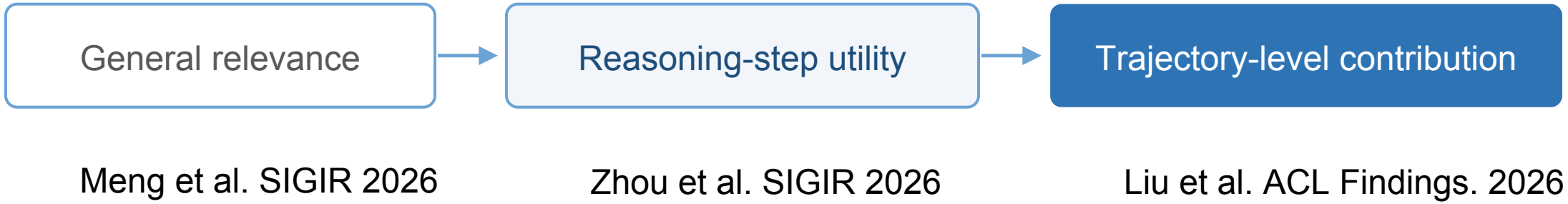
The improved agent yields better trajectories; **dual-signal labels** build its training data.



Liu et al. Agentic-R: Learning to Retrieve for Agentic Search. ACL Findings, 2026.

Document-side Utility — Recap

THE RETRIEVER OBJECTIVE SHIFTS



Document relevance ≠ document utility.

Return the evidence — at the right granularity — that best serves the current state.

The Three Utilities — Map

THE ACTION SPACE

retrieve? / browse / rewrite / verify / use tool / stop

The agent picks the action that best serves its need — including *whether/when to retrieve* — even bypassing the retriever.

Two questions: **which** action next, and **how** to reward it.

When to Retrieve

The first action-side call: retrieve now, or keep reasoning? Many state signals can trigger it.

SIGNAL OF NEED		OPERATIONALIZED BY
Low-confidence tokens	→	**FLARE** (Jiang et al., 2023)
Entropy · attention · stopwords	→	**DRAGIN** (Su et al., 2024)
Self-issued search actions	→	**Search-R1** (Jin et al., 2025)
Verifier / self-reflection failures	→	**Self-RAG** (Asai et al., 2024)
Contradictory evidence	→	**Conflicting Evidence** (Wang et al., 2024)

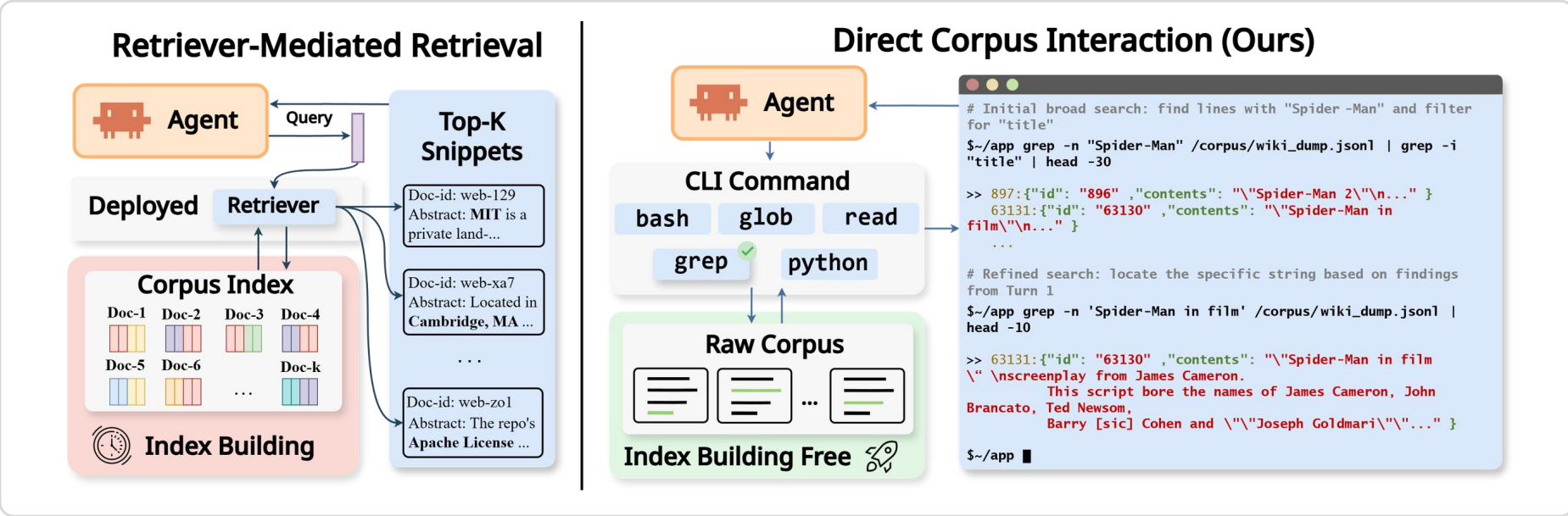
Direct Corpus Interaction

What about corpus access beyond retrieval?
an **action the agent chooses** — search raw files with **grep · rg · find**.

A fixed **top-k API** is just one hard-wired retrieval action. As agents get smarter, that interface — not their reasoning — becomes the bottleneck.

Li et al. Beyond Semantic Similarity: Retrieval for Agentic Search via Direct Corpus Interaction. arXiv, 2026.

Direct Corpus Interaction

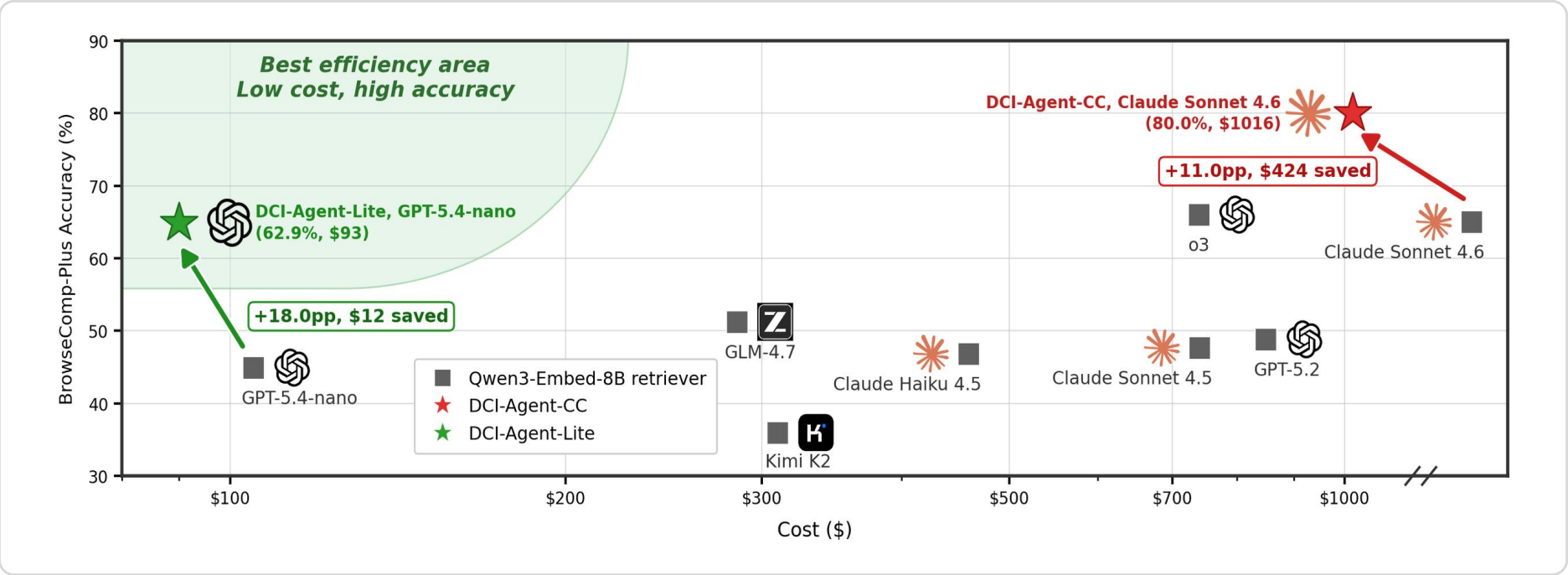


The far end of the action space — **bypass the retriever entirely**.

- **Zero-index** — no vector DB; adapts to new corpora instantly.
- **High-resolution control** — regex, joins, exact matches.

Li et al. Beyond Semantic Similarity: Retrieval for Agentic Search via Direct Corpus Interaction. arXiv, 2026.

Direct Corpus Interaction



Higher efficiency; higher accuracy

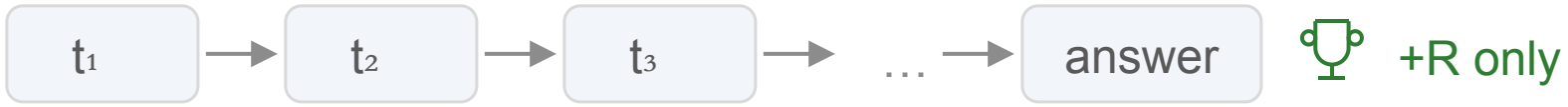
We've won 1st place on the multi-table QA subtask of NTCIR DAGRI task based on DCI!

Li et al. Beyond Semantic Similarity: Retrieval for Agentic Search via Direct Corpus Interaction. arXiv, 2026.

Intermediate Rewards

If the agent chooses among many actions, how does it learn which is right at each step?

Final-answer reward is sparse



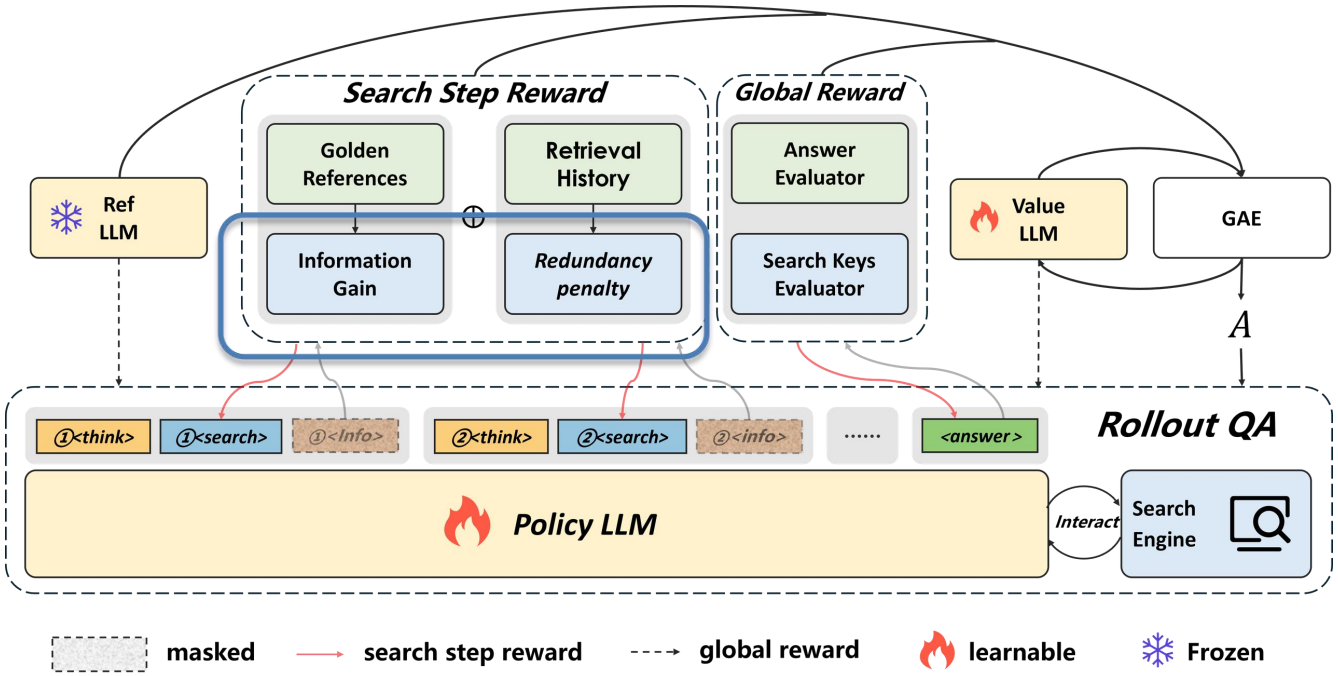
Intermediate rewards — dense per-step feedback

Translate utility metrics into intermediate rewards to guide step-by-step learning.

StepSearch

StepSearch rewards **each search step**, not just the final answer.

Global Reward = answer F1 + query alignment with ground-truth subqueries.
Step Reward = information gain − redundancy penalty.

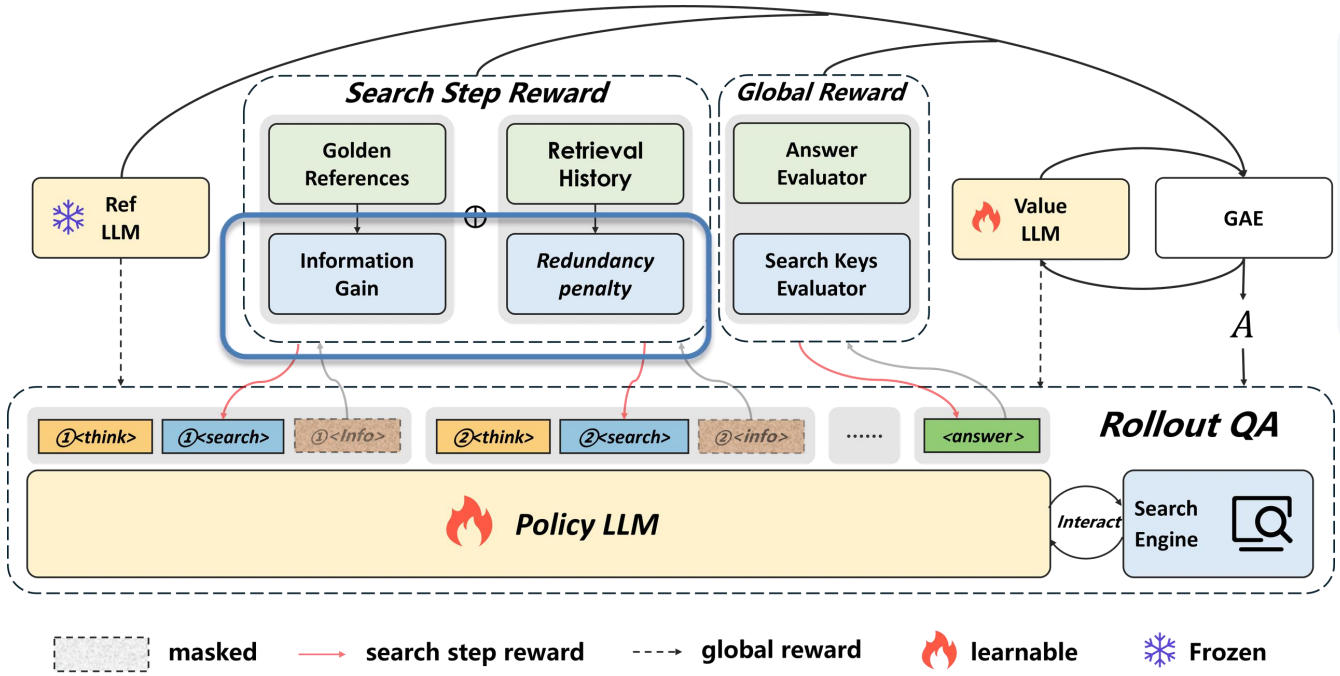


Wang et al. StepSearch: Igniting LLMs Search Ability via Step-Wise Proximal Policy Optimization. EMNLP, 2025.

StepSearch

StepSearch rewards **each search step**, not just the final answer.

Global Reward = answer F1 + query alignment with ground-truth subqueries.
Step Reward = information gain − redundancy penalty.



Information gain

- average marginal gain compared to the best historical match over n ground-truth docs

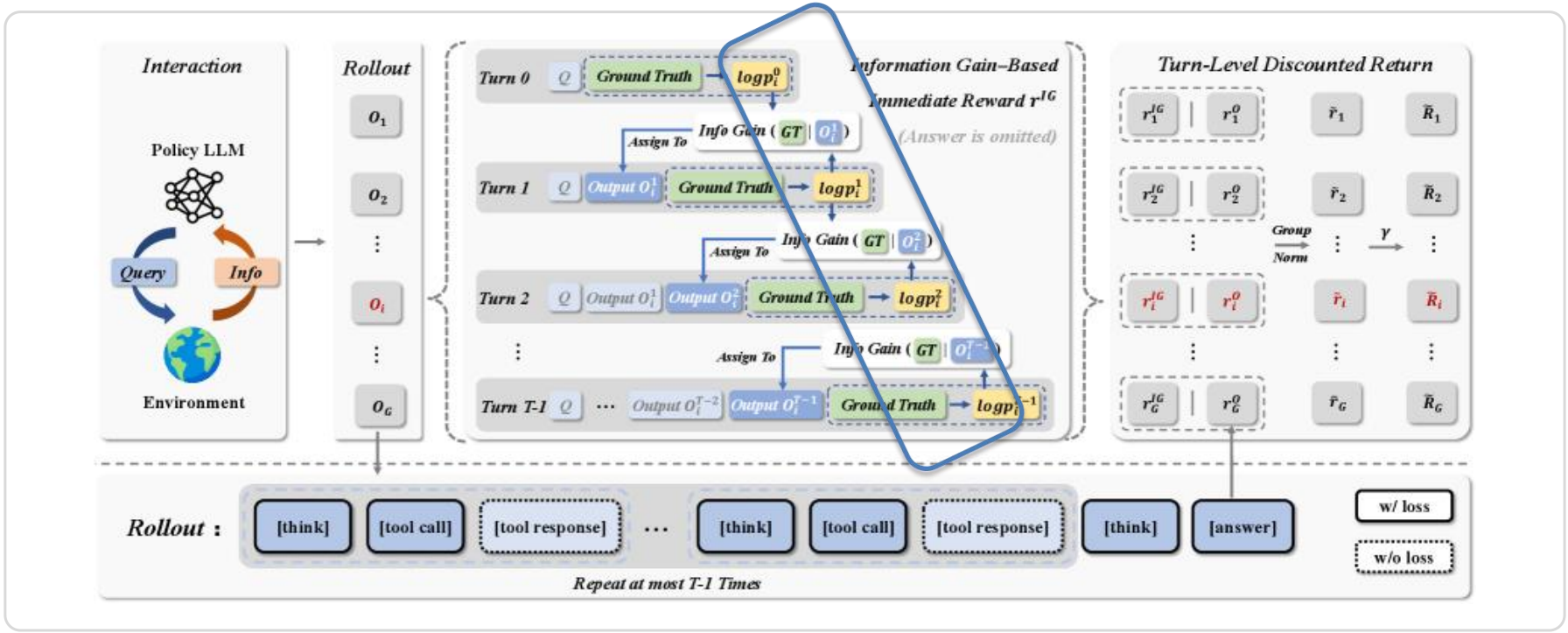
Redundancy Penalty

- fraction of retrieved docs already in the history

Wang et al. StepSearch: Igniting LLMs Search Ability via Step-Wise Proximal Policy Optimization. EMNLP, 2025.

IGPO — Belief Gain per Turn

IGPO rewards each turn by the **marginal rise** in the policy's own probability of the **ground-truth answer** after search.



Wang et al. Information Gain-based Policy Optimization: A Simple and Effective Approach for Multi-Turn LLM Agents. ICLR, 2026.

IGPO

Turn-level reward = the model's own **belief gain** about the ground-truth answer.

1. Log-probability of the gold answer under the current policy

2. **Info-gain reward** = marginal change in this probability after observing the tool (search) response

3. Group-normalized turn reward — info gain reward for $t < T$, outcome reward at $t = T$

Wang et al. Information Gain-based Policy Optimization: A Simple and Effective Approach for Multi-Turn LLM Agents. ICLR, 2026.

Action-side Utility — Recap

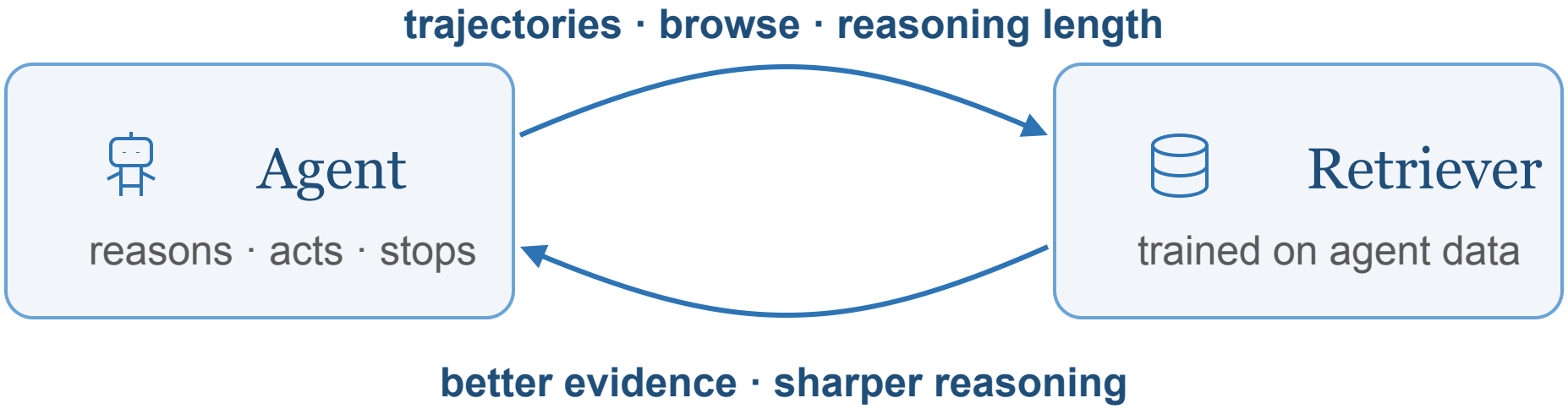
Action utility answers which action next — and how to reward it.

- choose the action that **maximizes progress** under the current state
- reward **marginal utility per step**, not just the final answer
- let **credit reach every intermediate decision**

> Sparse outcome reward → **dense, utility-shaped intermediate rewards.**

Retriever Adaptation

The retriever is not a frozen black box behind the agent.



The agent **shapes** the retriever; the retriever **sharpens** the agent.

The Three Utilities — Map

THREE PER-STATE UTILITIES

- **Query utility** — issue a state-relevant query
- **Document utility** — support the current step and the final answer
- **Action utility** — when to retrieve / browse / verify / stop / use tool

Retrieval should be optimized for trajectory utility, not only topical match.

Section 4 — Takeaways

1 **Agentic RAG makes utility state-dependent.**
Three per-state utilities — query, document, action — bound by a trajectory-level credit axis.

2 **Information need is dynamic.**
Signals decide when to retrieve (action) and what query to issue (query).

3 **Document utility is local and global.**
A passage relevant now can still hurt the final answer.

4 **Intermediate rewards turn utility into step-level signals.**
And retrievers should be trained from agent data, not human relevance.

NEXT — SECTION 5

We followed utility across a trajectory.

What's still open?

synthesis → one unifying map → open problems

Tutorial Roadmap

180 min total

01 **Introduction & Foundations -Keping** — 40 MIN
Why retrieval objectives must change for LLM consumers

02 **What Is LLM-Centric Utility? -Keping** — 50 MIN
Task, model, context, and presentation dimensions

03 **Utility Modeling & Optimization –Hengran** — 40 MIN
Signals, labels, trainable components, and evaluation

04 **Utility in Agentic RAG -Keping** — 40 MIN
Utility in dynamic retrieval and agent loops

05 **Open Problems & Q&A -Keping** — 10 MIN
Open problems and discussion

Tutorial website