30th ANNIVERSARY
EMNLP 2025
Suzhou, China | 中国苏州
November 4-9 | 11月4日–9日

s3: You Don’t Need That Much Data to Train a Search Agent

Pengcheng Jiang, Xueqiang Xu, Jiacheng Lin, Jinfeng Xiao, Zifeng Wang, Jimeng Sun, Jiawei Han

University of Illinois Urbana-Champaign

Presenter: Pengcheng (Patrick) Jiang

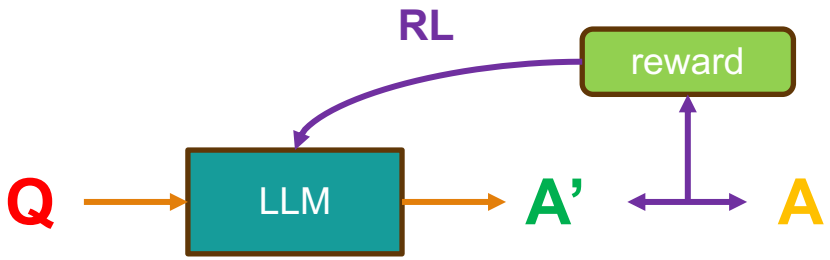
Overview

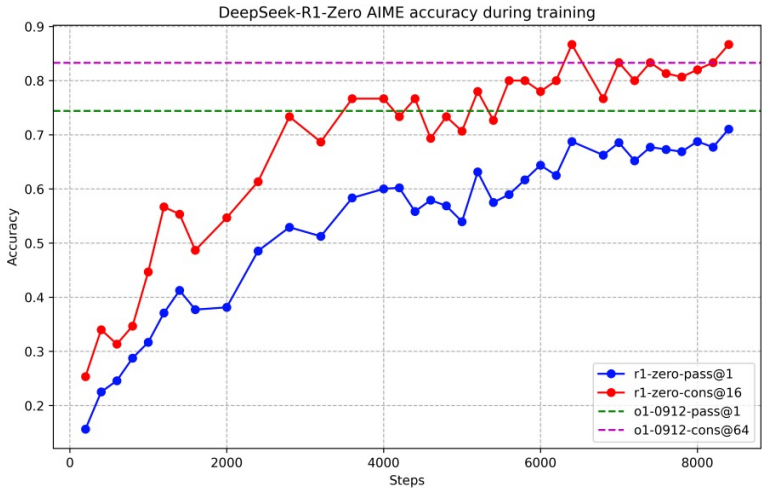
- Background
- Problems
- Method: s3
- Experiments & Takeaways
- Conclusion & Future Directions

Background

DeepSeek-R1-Zero[1] (*Nature,* 2025) demonstrates that LLMs can directly align with the task objective by Reinforcement Learning with Verifiable Reward (RLVR).

For example, for math problem-solving task:



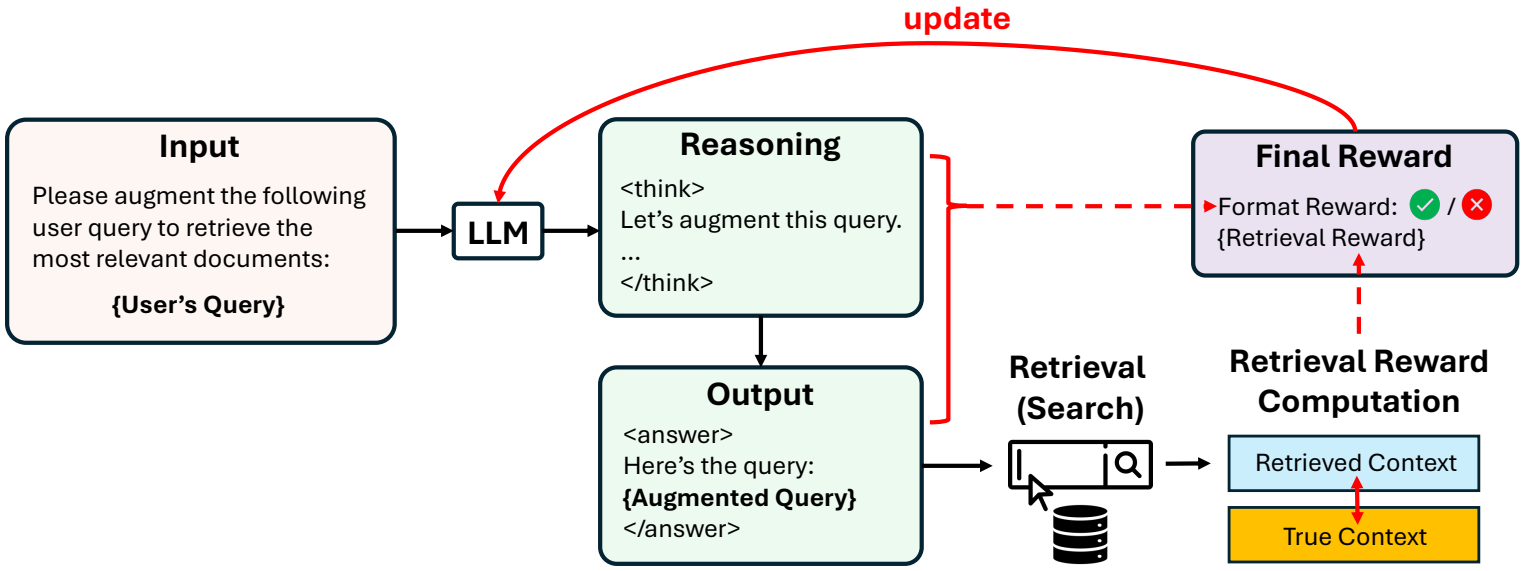


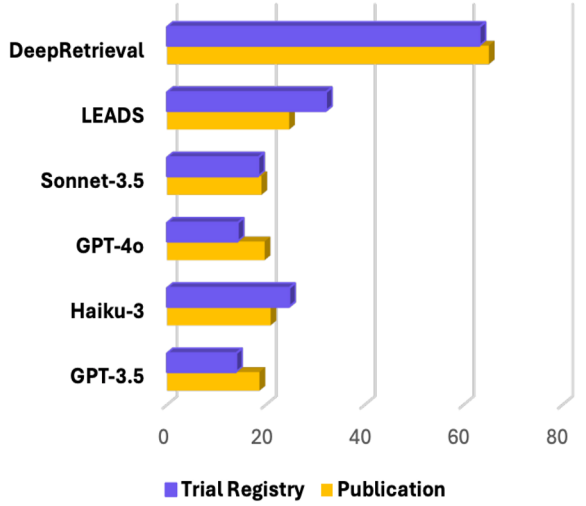
one can directly train an LLM with the numeric value (answer) match as the reward.

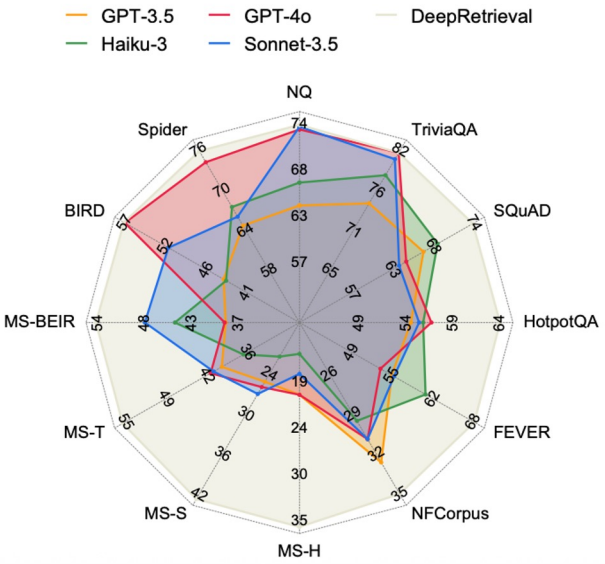
[1] Guo, et al. "Incentivizing reasoning capability in llms via reinforcement learning." *Nature*, 2025

Background

DeepRetrieval (COLM'25)





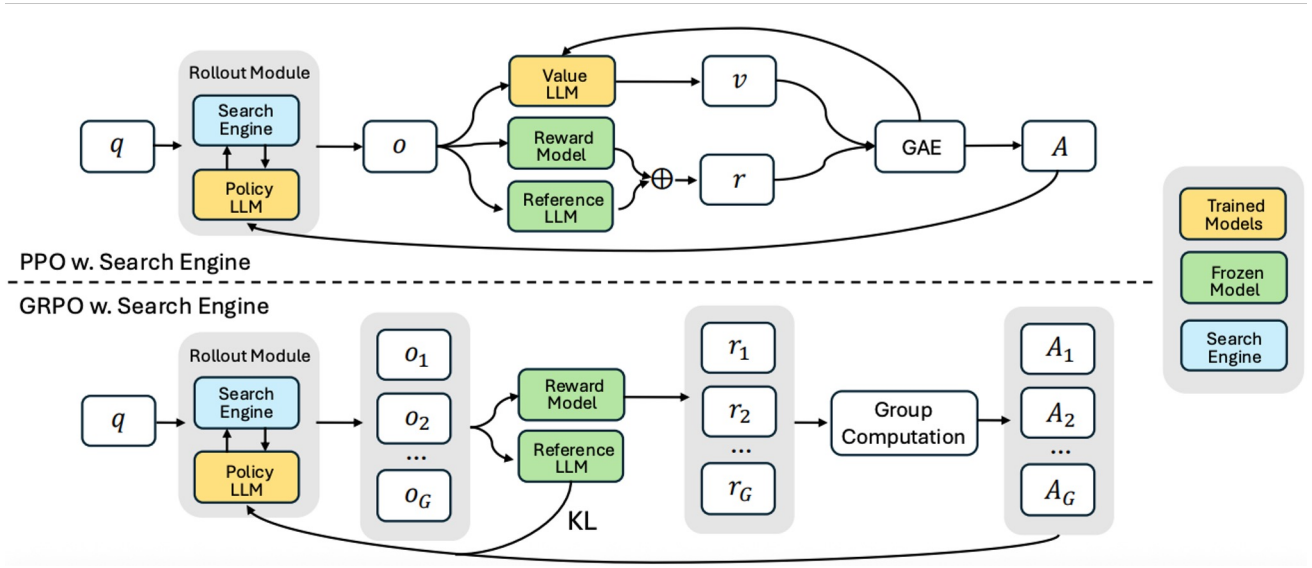


DeepRetrieval, focusing on the information retrieval task, uses **retrieval outcome** (e.g., recall, NDCG) as the signal to train a search agent with RL.

[2] P. Jiang , et al., "DeepRetrieval: Hacking Real Search Engines and Retrievers with Large Language Models via Reinforcement Learning," COLM'25

Background

Search-R1 (COLM'25)[3]



Algorithm 1 LLM Response Rollout with Multi-Turn Search Engine Calls

Require: Input query x, policy model π_θ, search engine $\mathcal{R}$, maximum action budget B.
Ensure: Final response y.

Initialize rollout sequence $y \leftarrow \emptyset$
Initialize action count $b \leftarrow 0$
while $b < B$ **do**
  Initialize current action LLM rollout sequence $y_b \leftarrow \emptyset$
  while True **do**
    Generate response token $y_t \sim \pi_\theta(\cdot \mid x, y + y_b)$
    Append y_t to rollout sequence $y_b \leftarrow y_b + y_t$
    if y_t in [</search>, </answer>, <eos>] **then** break
    end if
  end while
  $y \leftarrow y + y_b$
  if <search> </search> detected in y_b **then**
    Extract search query $q \leftarrow \text{Parse}(y_b, \text{<search>}, \text{</search>})$
    Retrieve search results $d = \mathcal{R}(q)$
    Insert d into rollout $y \leftarrow y +$ <information>d</information>
  else if <answer> </answer> detected in y_b **then**
    return final generated response y
  else
    Ask for rethink $y \leftarrow y +$ "My action is not correct. Let me rethink."
  end if
  Increment action count $b \leftarrow b + 1$
end while
return final generated response y

Search-R1, focusing on the RAG task, uses the **exact match (EM) of answer tokens** as the signal to train a search agent

Search-R1 trains the LLM to interact with retriever and reason over the searched context and generate an answer.

[3] B. Jin, et al., "Search-R1: Training LLMs to Reason and Leverage Search Engines with Reinforcement Learning," COLM'25

Problems

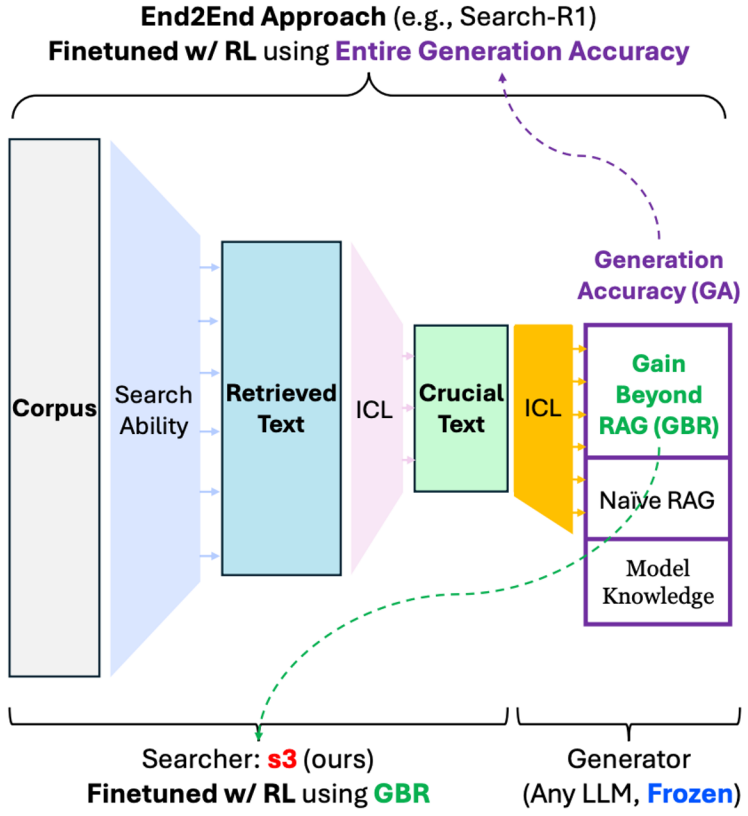
① Search-R1 is trained with exact match, which is **brittle**.

Why Exact Match Falls Short - An Example

Golden answer: "Barack Obama"
LLM response: "The 44th President of the United States was Barack Obama."
Exact match: 0 (response ≠ golden)
Generation Accuracy: 1 (span_check succeeds)

Therefore, EM is inappropriate to be seen as a "verifiable" reward for open QA task.

Cause: LLM will spend more training resources on aligning with answer tokens, making the contribution of search to the generation ambiguous.

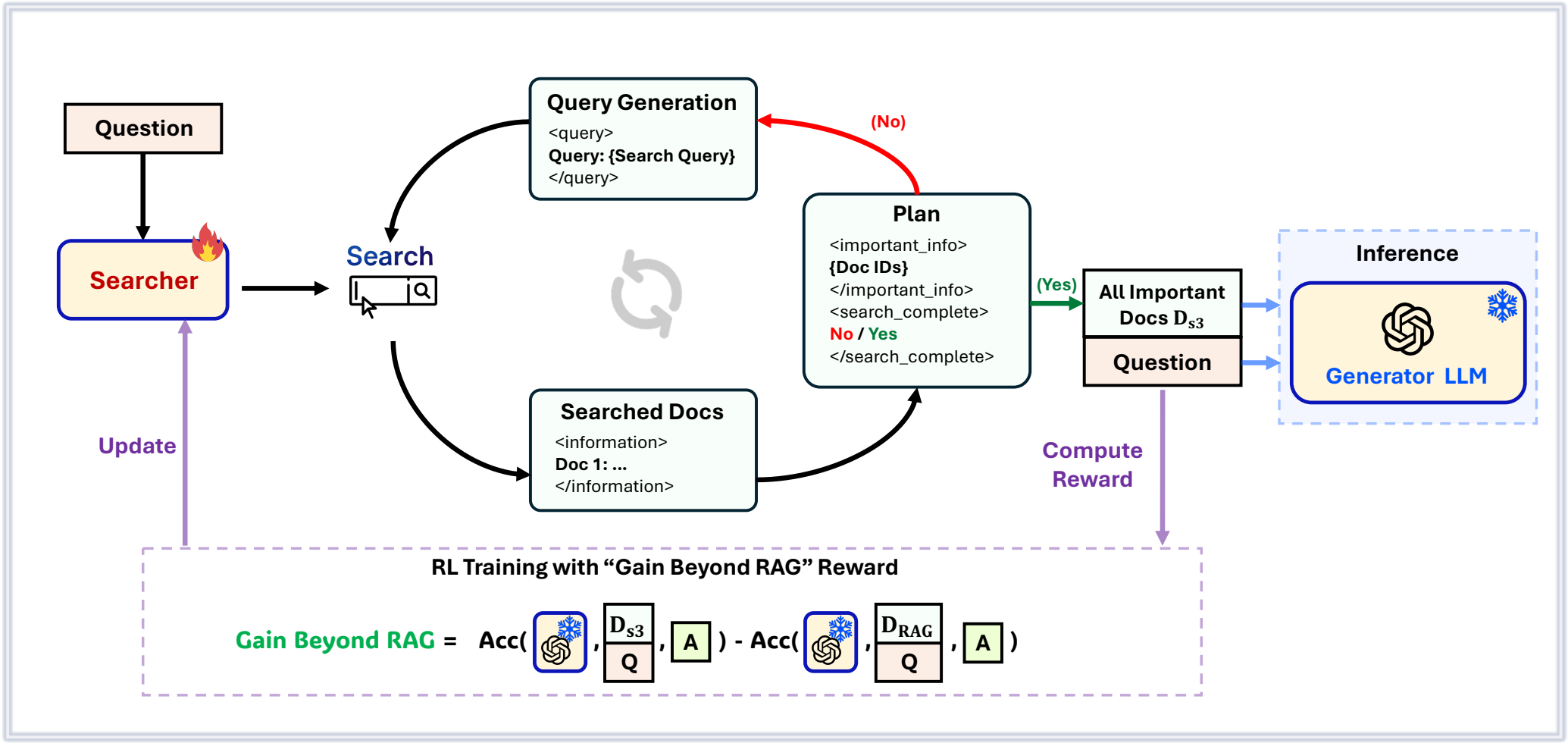
② 

Correct Generation can be achieved by:

- LLM's own knowledge
- Naïve RAG
- **Gain Beyond**

An effective reward that truly reflects the improved search quality
- How to measure?

Method: s3



: trained with RL : frozen D_{s3} D_{RAG} : searched docs by s3 / naïve RAG Q : question A : golden answer

Method: s3

s3 Loop

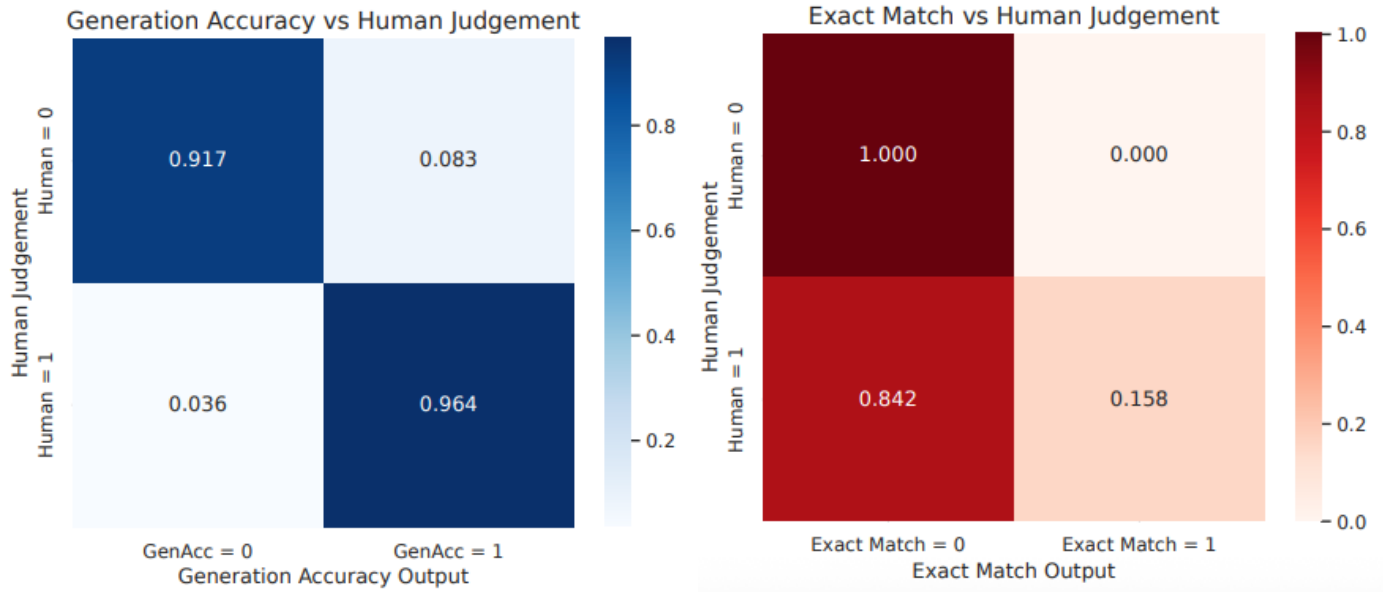
1. **Query Generation:** The searcher emits a query q_t in <query>...</query>.
2. **Search:** Documents $\mathcal{D}_t = \mathcal{R}(q_t)$ are retrieved in <information>...</information>
3. **Select:** Useful documents are selected between <important_info>...</important_info>, corresponding to subset $\mathcal{D}_t^{\text{sel}} \subseteq \mathcal{D}_t$.
4. **Stop decision:** The model declares <search_complete>[1/0]</search_complete>.

(For the first iteration, we start from search with the original query)

Evaluation Flow of Generation Accuracy

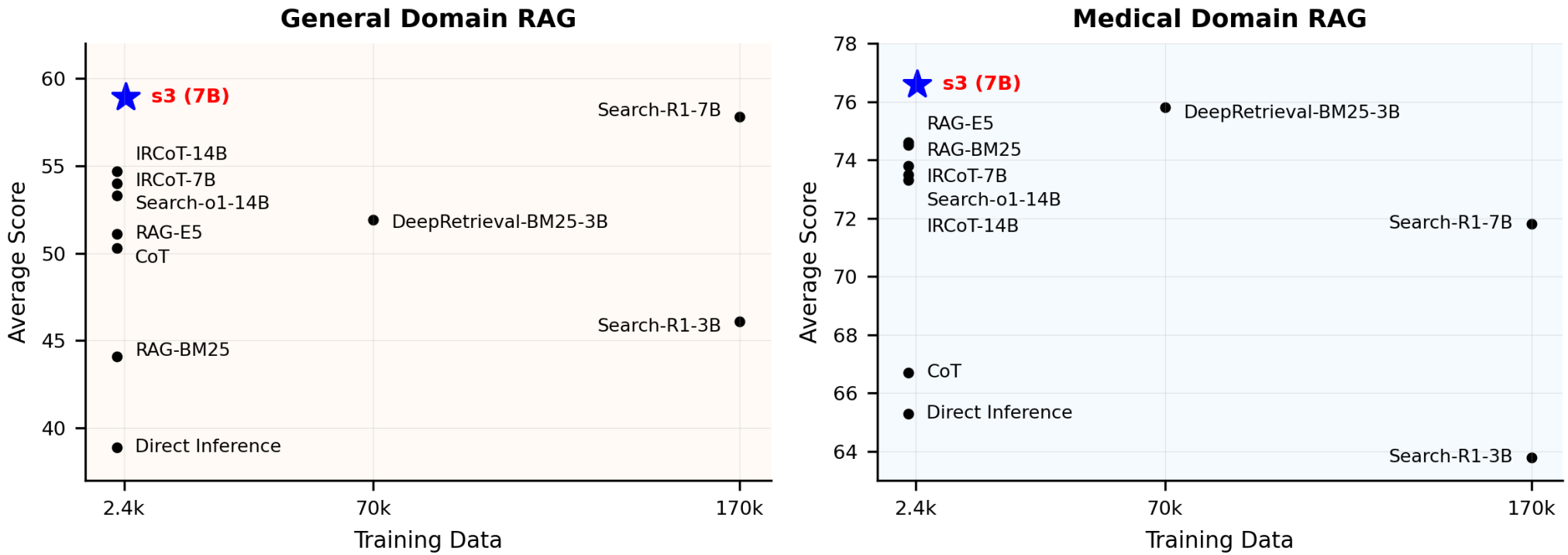
Input: Prediction p, Gold Answers $\mathcal{A}$

Step 1: Normalize p and $\mathcal{A}$ (lowercase, remove punctuation and articles).
Step 2: span_check $\rightarrow$ If any $a \in \mathcal{A}$ is a token span in p, return GenAcc = 1.
Step 3: judge_check $\rightarrow$ Prompt LLM: *"Does p contain any of $\mathcal{A}$?"*
Step 4: Return GenAcc = 1 if LLM says yes; else 0.



GenAcc is much more aligned with human judgement than exact match

Experiments & Takeaways



1. s3 outperforms all previous methods with **70x less training data** than Search-R1.

2. It also shows its **robustness to domain transfer** (from general to medical), without training on medical data.

Experiments & Takeaways

Methods	Searcher	#Train	Single-Hop			Multi-Hop			Avg.
			NQ†	TriviaQA	PopQA	HotpotQA†	2wiki	Musique	
#Test Data			3,610	11,313	14,267	7,405	12,576	2,417	
End-to-End Fine-Tuning									
$SFT_{Qwen2.5-3B-Inst}$	-	170k	23.7 (17.5)	41.6 (34.3)	18.1 (14.0)	18.0 (13.7)	22.1 (20.8)	5.1 (2.9)	21.4 (17.2)
$R1_{Qwen2.5-7B-Inst}$	-	170k	35.6 (28.8)	60.2 (53.4)	22.4 (20.5)	29.4 (24.0)	30.0 (29.1)	10.7 (7.8)	31.4 (27.3)
Search-R1-3B	(self) 3B	170k	47.0 (27.9)	65.6 (46.2)	46.4 (34.9)	33.5 (22.1)	28.5 (24.4)	6.0 (2.8)	37.8 (26.4)
Search-R1-7B	(self) 7B	170k	56.9 (48.2)	73.8 (64.0)	50.6 (46.8)	54.6 (43.5)	51.6 (38.4)	28.5 (20.6)	52.7 (43.6)
Generator (Qwen2.5-7b-Instruct) Frozen									
Direct Inference	-	0	37.3 (4.4)	55.1 (32.9)	19.9 (8.3)	28.1 (7.6)	36.9 (9.1)	10.6 (1.2)	31.3 (10.6)
CoT	-	0	37.7 (10.3)	60.6 (35.4)	22.2 (11.3)	31.1 (13.4)	31.6 (18.9)	10.6 (4.2)	32.3 (15.6)
RAG_{BM25}	-	0	43.6 (3.8)	69.8 (29.7)	34.6 (12.4)	45.3 (15.1)	38.5 (10.3)	11.5 (1.5)	40.6 (12.1)
RAG_{E5}	-	0	62.1 (5.8)	74.5 (33.8)	54.5 (20.3)	46.6 (13.6)	40.1 (7.8)	13.0 (2.0)	48.5 (13.9)
IRCoT	(self) 7B	0	63.2 (6.2)	75.6 (34.3)	54.5 (19.3)	50.9 (15.4)	48.7 (9.6)	16.4 (2.5)	51.6 (14.5)
IRCoT	14B	0	63.9 (6.3)	75.5 (34.9)	55.5 (20.3)	52.5 (16.0)	47.4 (9.3)	17.2 (2.7)	52.0 (14.9)
Search-R1-3B (Ret)	3B	170k	56.6 (6.6)	68.6 (32.5)	49.4 (18.8)	41.5 (13.6)	33.2 (7.8)	12.1 (1.9)	43.6 (13.5)
Search-R1-7B (Ret)	7B	170k	61.3 (8.1)	73.7 (35.9)	51.9 (20.7)	58.6 (20.0)	50.8 (12.2)	**27.6 (7.1)**	54.0 (17.3)
s3	7B	2.4k	**66.1 (7.2)**	**78.5 (36.8)**	**57.4 (21.9)**	**59.0 (21.8)**	**51.6 (12.4)**	23.9 (6.1)	**56.1 (17.7)**
Generator (Qwen2.5-14b-Instruct) Frozen									
Direct Inference	-	0	38.8 (8.2)	62.7 (39.0)	24.5 (10.8)	30.2 (9.5)	38.6 (7.2)	12.5 (1.8)	34.5 (12.8)
CoT	-	0	40.5 (10.2)	66.2 (41.6)	24.6 (13.6)	32.9 (12.3)	33.2 (13.8)	12.6 (5.2)	35.0 (16.1)
RAG_{BM25}	-	0	54.8 (16.4)	76.7 (44.8)	41.5 (22.7)	50.4 (18.3)	49.9 (6.4)	17.7 (3.1)	48.5 (18.6)
RAG_{E5}	-	0	62.4 (18.7)	77.4 (50.7)	55.1 (34.0)	47.4 (20.9)	44.9 (10.1)	16.1 (3.3)	50.6 (23.0)
IRCoT	7B	0	63.0 (18.8)	77.7 (50.1)	56.3 (33.5)	50.7 (22.7)	53.2 (12.4)	17.5 (4.1)	53.1 (23.6)
IRCoT	(self) 14B	0	63.9 (19.2)	78.2 (51.7)	56.1 (33.8)	51.6 (23.7)	54.0 (12.0)	19.1 (5.2)	53.8 (24.3)
Search-R1-3B (Ret)	3B	170k	59.2 (16.5)	75.6 (47.4)	52.3 (30.3)	45.5 (18.3)	44.0 (8.3)	16.0 (2.9)	48.8 (20.6)
Search-R1-7B (Ret)	7B	170k	63.8 (18.0)	76.3 (49.5)	54.6 (33.3)	56.7 (25.3)	56.7 (11.0)	**30.2 (9.1)**	56.4 (24.4)
s3	7B	2.4k	**67.2 (18.3)**	**79.5 (48.9)**	**57.8 (35.7)**	**57.1 (23.3)**	**57.1 (11.6)**	26.7 (7.8)	**57.6 (24.3)**
Generator (Claude-3-Haiku) Frozen									
Direct Inference	-	0	48.1 (25.7)	76.5 (64.8)	35.7 (30.9)	35.5 (24.2)	28.9 (24.0)	8.8 (4.3)	38.9 (29.0)
CoT	-	0	61.5 (2.9)	81.0 (30.0)	43.2 (9.1)	48.8 (8.8)	46.2 (6.8)	21.2 (2.3)	50.3 (10.0)
RAG_{BM25}	-	0	50.5 (3.8)	75.5 (28.4)	35.9 (8.0)	50.2 (11.4)	40.7 (8.1)	11.8 (0.8)	44.1 (10.1)
$DeepRetrieval_{BM25}$	3B	70k	64.4 (3.7)	80.2 (23.2)	45.5 (8.2)	54.5 (10.2)	47.1 (8.0)	22.2 (1.7)	52.3 (8.1)
RAG_{E5}	-	0	66.5 (4.3)	80.7 (28.9)	55.7 (8.9)	50.7 (11.5)	39.2 (7.8)	14.0 (1.2)	51.1 (10.4)
IRCoT	7B	0	68.0 (4.2)	81.7 (29.3)	55.5 (8.9)	54.8 (11.7)	46.5 (8.1)	17.4 (1.6)	54.0 (10.6)
IRCoT	14B	0	68.3 (4.2)	81.6 (29.5)	56.1 (8.6)	55.5 (11.9)	47.7 (8.4)	18.9 (1.7)	54.7 (10.7)
Search-o1	14B	0	67.3 (4.7)	81.2 (29.8)	50.2 (9.3)	58.1 (12.6)	48.8 (8.4)	14.2 (1.2)	53.3 (11.0)
Search-R1-3B (Ret)	3B	170k	60.7 (3.3)	74.5 (24.8)	50.1 (6.9)	45.7 (10.0)	33.1 (7.0)	12.7 (1.3)	46.1 (8.9)
Search-R1-7B (Ret)	7B	170k	68.1 (4.1)	80.9 (25.9)	55.7 (7.0)	62.0 (11.2)	51.0 (7.2)	**29.3 (3.2)**	57.8 (9.8)
s3	7B	2.4k	**70.5 (3.2)**	**84.0 (24.6)**	**57.7 (5.9)**	**62.4 (11.1)**	**52.4 (8.3)**	26.2 (7.9)	**58.9 (10.2)**

Methods	Searcher	#Train	Medical RAG-QA Datasets (MIRAGE)					Avg.
			MedQA-US	MedMCQA	PubMedQA	BioASQ-Y/N	MMLU-Med	
#Test Data			1,273	4,183	500	618	1,089	
w/o retrieval	-	0	61.7 (45.8)	55.8 (29.3)	55.6 (0.0)	76.9 (0.0)	76.4 (35.8)	65.3 (22.2)
Corpus: Wikipedia 2018 (Karpukhin et al., 2020)								
RAG_{BM25}	-	0	61.6 (48.2)	57.5 (45.2)	52.8 (4.6)	73.6 (6.3)	77.6 (61.9)	64.6 (33.2)
$DeepRetrieval_{BM25}$	3B	70k	62.5 (45.4)	61.3 (44.8)	56.2 (8.2)	**77.3 (9.2)**	**79.2 (57.9)**	67.3 (33.1)
RAG_{E5}	-	0	61.5 (46.7)	58.0 (44.7)	54.6 (3.8)	73.3 (5.3)	77.9 (62.2)	65.1 (32.5)
IRCoT	7B	0	62.8 (45.1)	60.5 (45.4)	54.2 (8.6)	73.0 (13.8)	78.7 (58.2)	65.8 (34.2)
IRCoT	14B	0	61.7 (48.9)	60.3 (46.7)	53.0 (7.6)	75.2 (11.8)	77.2 (61.9)	65.5 (35.4)
Search-o1	14B	0	64.5 (55.4)	59.6 (47.7)	52.2 (1.8)	74.9 (0.2)	77.7 (63.9)	65.8 (33.8)
Search-R1-3B (Ret)	3B	170k	58.8 (47.2)	53.7 (41.4)	53.8 (4.4)	63.6 (4.4)	68.4 (55.4)	59.7 (30.6)
Search-R1-7B (Ret)	7B	170k	62.6 (45.7)	59.2 (42.8)	55.4 (5.2)	71.2 (6.5)	69.3 (53.3)	63.5 (30.7)
s3	7B	2.4k	**65.7 (47.1)**	**61.5 (44.3)**	**56.6 (5.2)**	**77.3 (7.1)**	76.0 (56.3)	**68.3 (32.0)**
Corpus: Wikipedia+PubMed+Textbook (Xiong et al., 2024)								
RAG_{BM25}	-	0	65.4 (43.1)	59.9 (44.4)	79.4 (10.8)	88.4 (6.5)	79.6 (57.1)	74.5 (32.4)
$DeepRetrieval_{BM25}$	3B	70k	65.0 (35.1)	65.1 (44.2)	78.6 (16.2)	89.5 (7.4)	79.3 (49.1)	75.8 (30.4)
RAG_{E5}	-	0	64.1 (43.4)	60.1 (45.0)	79.4 (10.8)	89.8 (5.0)	78.8 (58.8)	74.6 (32.6)
IRCoT	7B	0	63.9 (38.6)	62.7 (45.3)	75.4 (13.0)	87.2 (5.8)	**79.7 (54.9)**	73.8 (31.5)
IRCoT	14B	0	62.7 (43.8)	62.3 (46.6)	74.0 (10.8)	87.9 (5.3)	79.6 (59.0)	73.3 (33.1)
Search-o1	14B	0	65.0 (50.1)	61.1 (47.6)	74.2 (12.0)	89.3 (5.3)	78.1 (59.5)	73.5 (34.1)
Search-R1-3B (Ret)	3B	170k	57.5 (45.5)	54.8 (40.7)	71.4 (7.8)	73.3 (3.6)	62.0 (47.6)	63.8 (29.0)
Search-R1-7B (Ret)	7B	170k	62.1 (43.2)	61.9 (44.2)	78.6 (8.0)	86.3 (5.3)	69.9 (48.9)	71.8 (29.9)
s3	7B	2.4k	**65.7 (45.7)**	**65.3 (45.4)**	**81.5 (13.6)**	**92.1 (6.5)**	78.3 (56.2)	**76.6 (33.5)**

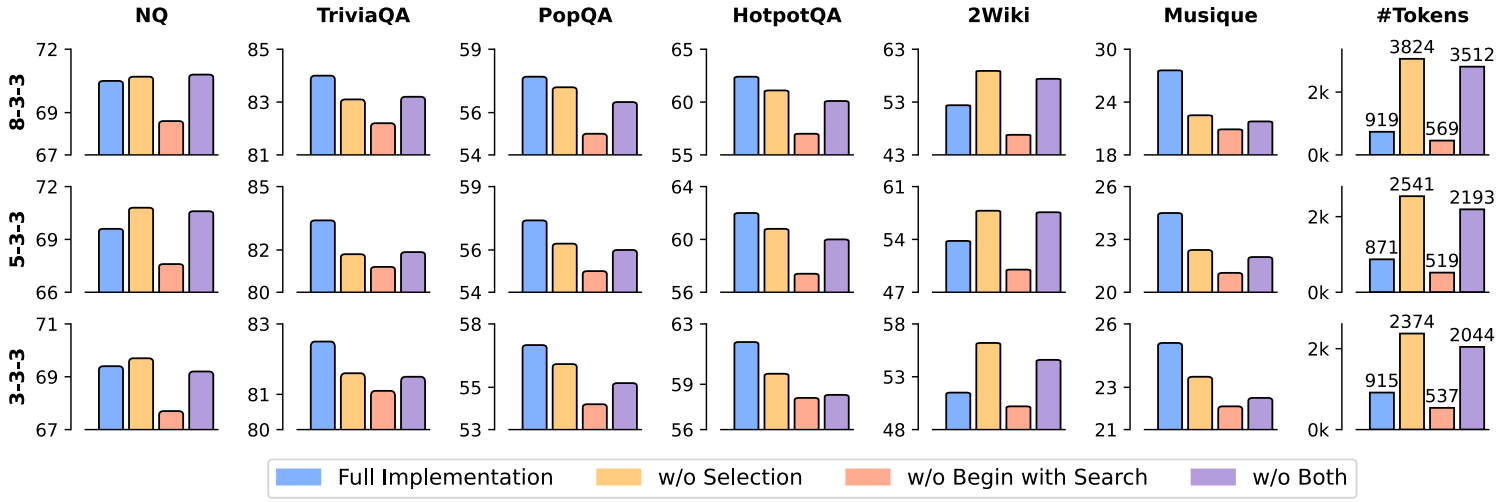
Takeaway #1: Searcher-Only is better than End-to-End Optimization for RAG

s3 consistently outperforms Search-R1 on search quality, revealing that most of the performance gain in RAG stems from improving the search capability instead of aligning generation outputs.

Takeaway #2: Searcher-Only Training enables Domain Transfer

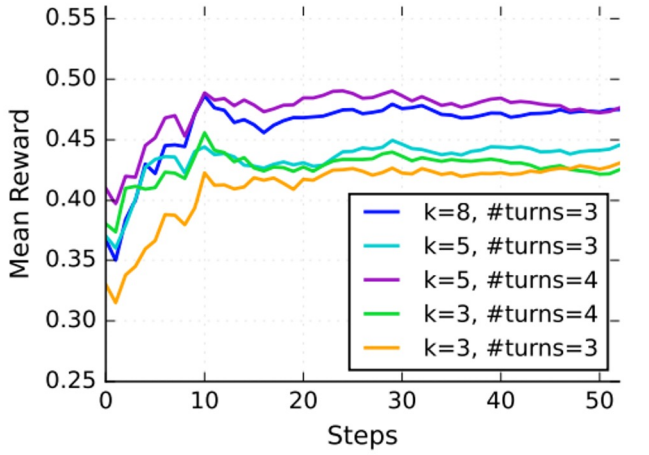
s3's zero-shot success on medical QA, despite training only on general QA, suggests that reinforcement-learned search skills generalize more reliably than generation-tuned approaches.

Experiments & Takeaways



Findings:

- **Begin with search** by original question is important, as it can avoid the trajectory deviate from the original query intent.
- **Selection** process can effectively reduce the token consumption, without compromising the overall performance.
- EM is a brittle metric to evaluate LLM's generation for open QA.



	GenAcc	LLMJudge	Span	EM
General QA	58.9	59.6	57.1	50.5
Medical QA	76.6	77.3	74.3	70.3

Takeaway #3: Reward Choice directly shapes Search Quality

Using semantically or human preference aligned metrics like our GenAcc (§4.1) encourages the search policy to retrieve substantively helpful documents, rather than optimizing for brittle string overlap.

Conclusion & Future Directions

We present s3, a framework that

- Trains a search-only agent using the Gain Beyond RAG reward.

- By decoupling search from generation and optimizing only the retriever, s3 outperforms strong baselines with just 2.4k examples.

- Our results show that targeted search policy learning yields substantial gains in both efficiency and generalization, offering a scalable path for improving RAG systems.

Future Directions:

- s3 shows that we can efficiently train a task-specific auxiliary agent while keeping the main reasoning model frozen. This modular paradigm can scale to many other tasks.

- Although search and answering are decoupled, the answering stage remains unoptimized. A natural extension is to train a lightweight answering-specific agent that reasons more effectively over the searched context.

Code: https://github.com/pat-jj/s3

Contact: Patrick (Pengcheng) Jiang, pj20@illinois.edu