PaperSearchQA: Learning to Search and Reason over Scientific Papers with RLVR

James Burgess[1*] **Jan N. Hansen**[1] **Duo Peng**[2] **Yuhui Zhang**[1] **Alejandro Lozano**[1]
Min Woo Sun[1] **Emma Lundberg**[1,2,3] **Serena Yeung-Levy**[1,2]
[1]Stanford University, [2]Chan Zuckerberg Biohub Network , [3]KTH Royal Institute of Technology
`https://github.com/jmhb0/PaperSearchQA`

Abstract

Search agents are language models (LMs) that reason and search knowledge bases (or the web) to answer questions; recent methods supervise only the final answer accuracy using reinforcement learning with verifiable rewards (RLVR). Most RLVR search agents tackle general-domain QA, which limits their relevance to technical AI systems in science, engineering, and medicine. In this work we propose training agents to search and reason over scientific papers – this tests technical question-answering, it is directly relevant to real scientists, and the capabilities will be crucial to future AI Scientist systems. Concretely, we release a search corpus of 16 million biomedical paper abstracts and construct a challenging factoid QA dataset called PaperSearchQA with 60k samples answerable from the corpus, along with benchmarks. We train search agents in this environment to outperform non-RL retrieval baselines; we also perform further quantitative analysis and observe interesting agent behaviors like planning, reasoning, and self-verification. Our corpus, datasets, and benchmarks are usable with the popular Search-R1 codebase for RLVR training; they are available on Hugging Face[1]. Finally, our data creation methods are scalable and easily extendable to other scientific domains.

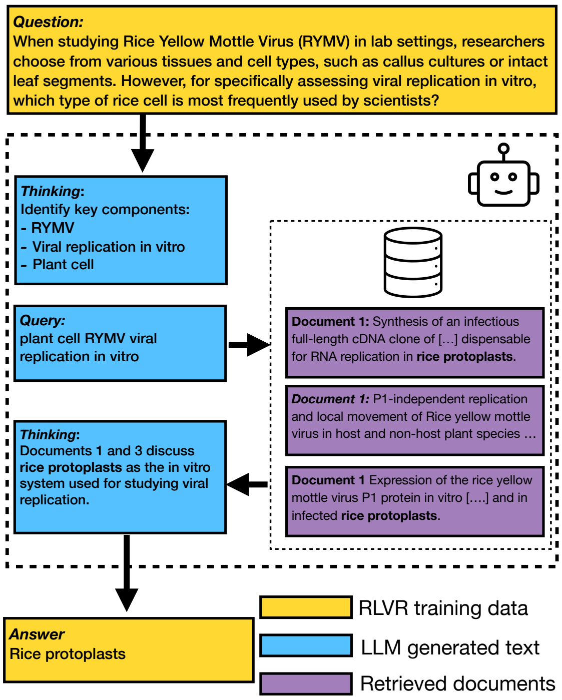

Question:
When studying Rice Yellow Mottle Virus (RYMV) in lab settings, researchers choose from various tissues and cell types, such as callus cultures or intact leaf segments. However, for specifically assessing viral replication in vitro, which type of rice cell is most frequently used by scientists?
Thinking:
Identify key components:
- RYMV
- Viral replication in vitro
- Plant cell
Query:
plant cell RYMV viral replication in vitro
Document 1: Synthesis of an infectious full-length cDNA clone of [...] dispensable for RNA replication in **rice protoplasts**.
Document 1: P1-independent replication and local movement of Rice yellow mottle virus in host and non-host plant species ...
Document 1 Expression of the rice yellow mottle virus P1 protein in vitro [....] and in infected **rice protoplasts**.
Thinking:
Documents 1 and 3 discuss **rice protoplasts** as the in vitro system used for studying viral replication.
Answer
Rice protoplasts
RLVR training data
LLM generated text
Retrieved documents

Figure 1: Search agents interleave reasoning and retrieval for question answering (QA). We study QA over scientific literature, contributing an environment for training agents with RL with verifiable rewards (RLVR). We release a training dataset of factoid QA (yellow boxes), a retrieval corpus (purple), and benchmarks.

1 Introduction

Following the release of Deepseek-R1 (Guo et al., 2025) and OpenAI's o1 (Jaech et al., 2024), much large language model (LLM) research has employed reinforcement learning with verifiable rewards (RLVR) (Shao et al., 2024; Lambert et al., 2024). In RLVR, an LLM is prompted to answer a query, and a reward is given only if an automatic verifier deems the final output correct; the corresponding tokens are then used to update the model (appendix D has more details). This differs from supervised finetuning (SFT), which learns directly from labeled text sequences. Early follow-up work focused on math and code applications (Chen et al., 2025), followed by tool-use agents (Feng et al., 2025) where the LLM both calls tools and reasons over their outputs to complete tasks. Compared to earlier ways of controlling agents such as prompting, scaffolding, and supervised finetuning, RLVR is appealing for its potential to incentivize more general and flexible behavior and reasoning (Chu et al., 2025; Guo et al., 2025).

One major application of tool-use LLMs is

[*] Correspondence: `jmhb@stanford.edu`

[1] https://huggingface.co/PaperSearchQA/datasets

knowledge-intensive question-answering. Here, *search agents* can reason about the query and search over knowledge bases (KBs) in an interleaved fashion (Yao et al., 2023; Trivedi et al., 2022; Li et al., 2025a). RLVR was shown to be effective for training search agents by Search-R1 (Jin et al., 2025), along with many concurrent and follow-up papers (Song et al., 2025; Sun et al., 2025; Zheng et al., 2025). However these works emphasize general-knowledge QA that test simple trivia (Kwiatkowski et al., 2019; Yang et al., 2018; Joshi et al., 2017; Ho et al., 2020), and not technical and knowledge-intensive domains like science, engineering, law, and medicine. These require more technical knowledge, reasoning about complex systems, and ability to search technical knowledge bases.

One promising setting for training technical reinforcement learning (RL) search agents is in scientific AI systems (Lu et al., 2024; Gao et al., 2024). Scientific research has a huge volume of knowledge in databases and literature (Ferguson et al., 2014; Delile et al., 2024), and traversing that knowledge is an essential part of every stage of the research process (Hope et al., 2023). The interest in AI search has been established by literature retrieval systems (Lála et al., 2023; Asai et al., 2024), and we predict that future complex agent systems for AI research will include modules for searching scientific literature and knowledge bases (Lu et al., 2024). These search modules will require specialist domain understanding to properly perform query formulation, to reason about retrieved information, and to evaluate the quality of the retrieved information.

In this work, we propose training RL search agents to search and reason over a corpus of research papers to answer scientific questions. We focus on easily-verified factoid questions, for example *What gene is mutated in childhood retinoblastoma?* (Answer *RB1*); such queries are amenable to current RLVR training, while also being useful to practicing scientists (Krithara et al., 2023). Specifically, we first release a corpus and search index of 16 million abstracts from biomedical papers in PubMed. Second, we release a dataset of 60k factoid QAs; the datasets are generated from the Pubmed articles in an LLM workflow, that underwent rigorous quality assurance by biology experts for correctness and relevance to a real scientific search application. The data creation methods are highly scalable, and can be adapted to other domains like materials science or chemistry. Third, and for benchmarks, we reserve 5k samples for testing, and we re-distribute the factoid subset of BioASQ, a small scale but high quality human-created dataset (Krithara et al., 2023).

We train LLM search agents in our environment, showing that current RL training techniques (Jin et al., 2025) lead to stronger performance compared to non-RL baselines. However the overall scores remain low, which establish our datasets as challenging for training search systems. We perform quantitative analysis, finding: general-domain semantic retrievers offer small benefits compared to syntactic retrievers; LLMs without retrievers have non-negligible performance; gains to accuracy with model size are likely due to better parametric knowledge; and paraphrasing in dataset construction adds dataset difficulty. Additionally, our qualitative results show interesting behaviors, specifically simple planning about query rewriting, reasoning about questions before retrieval, and verification when the model already has an initial answer.

In summary, our contributions are: - A new environment for training search agents in scientific question answering over papers: specifically a corpus, training datasets, and benchmarks. - Demonstrating successful RLVR training of search agents over scientific papers, with quantitative and qualitative insights.

2 Related Work

We review general-domain search agents, followed by systems for understanding scientific literature.

2.1 Search agents

Search-R1 (Jin et al., 2025) and R1-Searcher (Song et al., 2025) were the first open search agents for question answering trained using reinforcement learning with final-answer reward. (Closed systems like OpenAI's o3 (OpenAI, 2025b) and Deep Research likely explored this earlier (OpenAI, 2025a)). There were many followups exploring, for example, search in web environments (Zheng et al., 2025; Li et al., 2025b), query decomposition (Guan et al., 2025), and simulating the retrieval environment (Sun et al., 2025). We contribute to this direction by proposing new RL training environments; while prior works emphasize general knowledge QA, we create datasets, evaluations, and a retrieval corpus for training agents to reason over scientific literature. Earlier, search agents (and RAG systems) were supervised with supervised fine-tuning (Schick et al., 2023), few-shot prompting (Yao et al., 2023; Trivedi et al., 2022), or prompt optimization (Opsahl-Ong et al., 2024); these approaches likely lead to worse generalization (Chu et al., 2025; Guo et al., 2025). Concurrent with recent search agents, many train agents with RL for tool use beyond search engines (Feng et al., 2025; Qian et al., 2025).

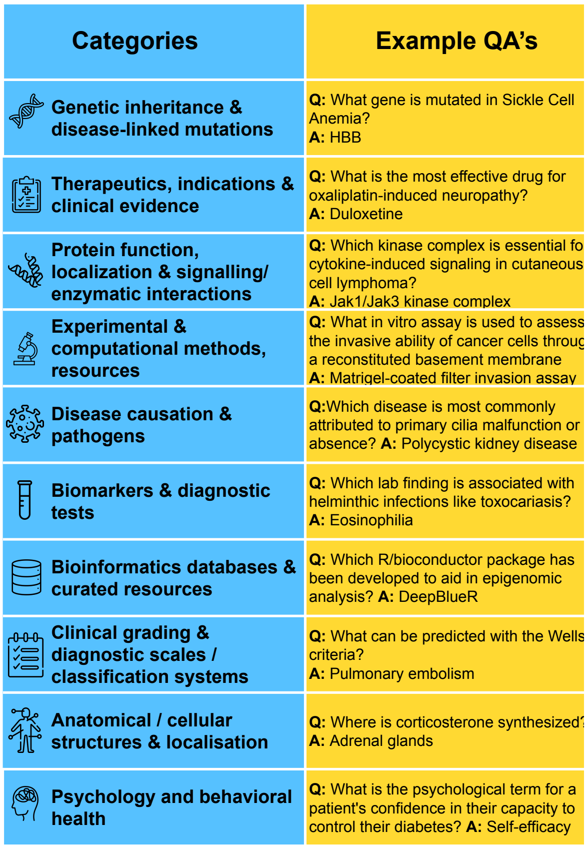
2.2 Search Agents for Scientific QA

BioASQ (Tsatsaronis et al., 2015; Krithara et al., 2023) is an annual challenge run since 2012 for benchmarking semantic indexing and open-domain question-answering for scientific literature – its popularity reflects the importance of literature understanding tasks for practicing scientists. Their task definitions influence our dataset construction, though a limitation is that their human-generated data is hard to scale. There are many systems for open-domain question-answering over literature, include PaperQA (Lála et al., 2023; Skarlinski et al., 2024) and OpenScholar (Asai et al., 2024). They have impressive capabilities, handling large corpora of full-text articles, however the agent behavior is controlled by component scaffolding, prompt engineering, or supervised fine-tuning. Instead, we explore training agents with RL because it promises stronger generalization in the long term (Chu et al., 2025; Jin et al., 2025). To make progress in this direction, we focus on *factoid* QA, where answers are easy to unambiguously verify. Note that current RL-trained search agents are designed for factoid QA (Jin et al., 2025), while such questions are useful to applications (Krithara et al., 2023). This motivates us generating new datasets, since prior datasets have binary answers (Jin et al., 2019; Wadden et al., 2020), have long-form answers with fuzzy evaluation (Asai et al., 2024; Lee et al., 2023), or they have smaller scale (Skarlinski et al., 2024).

3 Methods

In the following sections, we first describe the training data construction process, then the search corpus and indexing, and finally the RL training algorithm.

3.1 Dataset Construction



Categories	Example QA's
Genetic inheritance & disease-linked mutations	**Q:** What gene is mutated in Sickle Cell Anemia? **A:** HBB
Therapeutics, indications & clinical evidence	**Q:** What is the most effective drug for oxaliplatin-induced neuropathy? **A:** Duloxetine
Protein function, localization & signalling/ enzymatic interactions	**Q:** Which kinase complex is essential fo cytokine-induced signaling in cutaneous cell lymphoma? **A:** Jak1/Jak3 kinase complex
Experimental & computational methods, resources	**Q:** What in vitro assay is used to assess the invasive ability of cancer cells throug a reconstituted basement membrane **A:** Matrigel-coated filter invasion assay
Disease causation & pathogens	**Q:**Which disease is most commonly attributed to primary cilia malfunction or absence? **A:** Polycystic kidney disease
Biomarkers & diagnostic tests	**Q:** Which lab finding is associated with helminthic infections like toxocariasis? **A:** Eosinophilia
Bioinformatics databases & curated resources	**Q:** Which R/bioconductor package has been developed to aid in epigenomic analysis? **A:** DeepBlueR
Clinical grading & diagnostic scales / classification systems	**Q:** What can be predicted with the Wells criteria? **A:** Pulmonary embolism
Anatomical / cellular structures & localisation	**Q:** Where is corticosterone synthesized? **A:** Adrenal glands
Psychology and behavioral health	**Q:** What is the psychological term for a patient's confidence in their capacity to control their diabetes? **A:** Self-efficacy

Figure 2: Left: the ten question-answering categories defined with experts. Right: example question-answer pairs, which are sufficient supervision for RLVR training methods.

Defining Dataset Properties The first main goal is that question-answer pairs (QA's) can serve as training data for methods needing *outcome* supervision – for example, reinforcement learning with verifiable rewards (RLVR) (Jin et al., 2025). Specifically, the answers mut be *verifiable* – it should be possible for a reward model to judge whether the prediciton matches the ground truth answer without any ambiguity. To satisfy verifiability, we make the following design decisions. QA's are *factoid*, meaning the answer is a single entity; this is similar to the most popular general-knowledge QA datasets studied by search agents (Kwiatkowski et al., 2019; Joshi et al., 2017). (Alternative and more complex formulations, like 'list of entities' have been left to future work). The questions are unambiguous: written so that only a single entity name (or its synonyms) are correct. Then, the reward model is simply checking whether the prediction is equal to the ground truth answer (or its synonyms). We ensure questions have a low 'random guessing baseline', because this can lead to incorrect reasoning frequently being rewarded, which is noisy supervision. In particular, we do not allow binary answers (e.g. True or False), and our quality control process ensures that the question text rarely gives a small list of options. Another property – implicit in our construction pipeline – is that questions are single-hop, meaning they can be answered from a single correctly-retrieved document. Since we employ outcome-only reward, we do not require annotations for intermediate reasoning or for retrieved documents.

The second main goal is that QA's should be relevant to real applications: they must be questions that real scientists might ask in their work. To ensure this, our team includes practicing scientists at all stages – from defining task properties to pipeline construction to verifying the data. Our construction pipeline also take inspiration from the BioASQ project (Krithara et al., 2023; Nentidis et al., 2023; Tsatsaronis et al., 2015) – a challenge for semantic indexing and question-answering (including factoid-QA) over biomedical articles – that has run since 2015, and garnered significant attention in bioinformatics and NLP. While a limitation of BioASQ is that questions are human-created and therefore difficult to scale, it clearly demonstrates the significant interest in biomedical question answering over scientific papers; this supports our claim that PaperSearchQA is interesting to applications.

Categories for Question-Answering To ensure the QA-generation pipeline produces questions that satisfy our key target properties – unambiguous factoid and relevant to application – we defined ten target question categories. The categories and examples are shown in Figure 2.

To develop these, first the human experts on our team performed brainstorming to identify one candidate category set. Next, we sampled 300 questions from the BioASQ database and used LLMs (Claude Opus 4 (Anthropic, 2025) and OpenAI o3 (OpenAI, 2025b)) to propose two more candidate category sets. Then, the human experts synthesized those into a final list, which required some merging and discarding rare categories. The final category names with examples were used in the data construction pipeline.

Automatic QA Generation Pipeline The data generation process (Fig.Y) uses paper abstracts as a knowledge source, which are then mapped to QAs using an LLM workflow. The LLM prompts and pipeline architecture were iteratively designed based on expert review from biomedical scientists. Specifically we generate 200 questions, the expert provides text feedback; the human prompt engineer then modifies the workflow topology or the LLM instructions with metaprompting (Schulhoff et al., 2024).

First, the paper abstracts are randomly sampled from the corpus described in Section 3.3 – the same corpus that is searched at inference time. The abstract is passed to an LLM (GPT-4.1 (Achiam et al., 2023)) with a carefully-designed prompt (see Appendix E). This prompt includes the target categories from Figure 2, along with guidance ensuring the questions are suitable for open-domain QA: factoid answers, no acronyms, and no assumed access to the document (and we add an extra filtering step for phrases like 'this study'). We found that generating three questions per abstract led to better dataset diversity.

Since reward models commonly use exact match comparison of prediction and target, we generate synonyms of 'golden answers', using GPT-4.1 (prompt in Appendix E). Next, we notice that questions often use exact keywords and phrasing found in the abstract, while realistic use-cases would often use synonyms. We therefore sample 50% of QAs for question rewriting, and use an LLM prompt to 'paraphrase' the question with different terminology (prompt in Appendix E). Finally, dataset is split into train and test randomly.

All LLM calls were made through OpenRouter. The total cost, including experimentation and final data generation, was estimated at $600.

Dataset Summary The final PaperSearchQA-dataset has 54,907 training samples and 5,000 test samples. For question categories Figure 2, the top categories are 'Experimental & computational methods' (27%) and 'Therapeutics, indications & clinical evidence'. Median question word length is 18 and median answer word length is 2. Each sample is annotated with the Pubmed ID of the source paper, the category, and whether the question was paraphrased to avoid easy keyword matching. It is available on Hugging Face Hub and is released with a CC-BY license.

3.2 Evaluation dataset: BioASQ

BioASQ is a popular challenge for biomedical indexing and question answering, where all samples are human-creating (Krithara et al., 2023; Tsatsaronis et al., 2015). Due to it's smaller scale, we propose using it for search agent evaluation, where the search corpus is the same PubMed abstracts

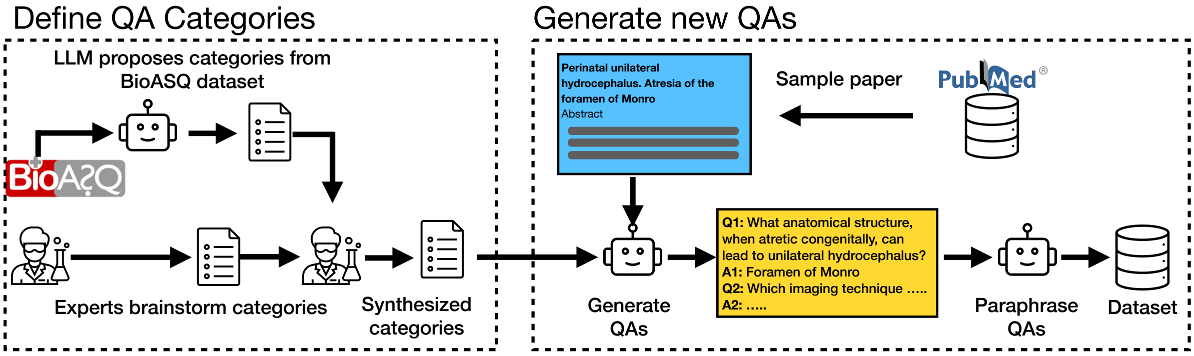

Define QA Categories
LLM proposes categories from BioASQ dataset
BioASQ
Experts brainstorm categories
Synthesized categories
Generate new QAs
Perinatal unilateral hydrocephalus. Atresia of the foramen of Monro
Abstract
Sample paper
PubMed
Q1: What anatomical structure, when atretic congenitally, can lead to unilateral hydrocephalus?
A1: Foramen of Monro
Q2: Which imaging technique .....
A2: .....
Generate QAs
Paraphrase QAs
Dataset

Figure 3: Data generation pipeline process. Left, generating the categories from Figure 2: LLM summarizes categories from human-written questions in BioASQ (Krithara et al., 2023); humans brainstorm categories in parallel; humans synthesize both sources into final categories. Right, QA generation: abstracts from PubMed are sampled and passed to an LLM. The LLM s prompted with categories (and other guidance) to generate QAs. A second LLM paraphrases the QAs to limit exact keyword matching.

from Section 3.3. For convenience, we collect data from all years up to 2025 and redistribute it on Huggingface Hub. Our only addition is to generate synonyms for the answer into the 'golden answer' list (using the same LLM call from our own pipeline) which enables exact-match evaluation metric. It is released under CC-BY-2.5 license. Its 'factoid' dataset has 1,609 samples. BioASQ has question categories other than *factoid* – *yes/no*, *list*, and *summary* – which we also release, though we do not use it in this paper.

3.3 Retrieval Corpus and Index

The search corpus is 16 million PubMed abstracts up to 2025, and was previously distributed by BioASQ (Krithara et al., 2023)[2]. We concatenate the paper title with the abstract text, giving a mean word length of 245.

We provide BM25 (Robertson and Walker, 1994) and e5 (Wang et al., 2022) search indexes. The corpus and index is small enough to hold in memory: the corpus is 23GB, the BM25 index is 2.6GB, and the e5 index is 93GB. At inference time, the e5 retriever index requires two A100s GPUs (80GB) to avoid memory error at inference.

3.4 Training Algorithms

To demonstrate the value of our datasets and retriever, we train search agents using RLVR.

RLVR for Search Agents We follow Search-R1 (Jin et al., 2025), which uses reinforcement learning with verifiable rewards (RLVR).

We provide a minimal system prompt (Appendix F), which introduces the question-answering task, instructing the model to leverage reasoning tokens inside `<think>` tokens and to give the final answer inside `<answer>` tokens. The prompt then describes usage of search: by wrapping queries in `<query>` tokens. When a query is found, the system stops generation, extracts the query, and retrieves the top k documents. It then appends the documents to the reasoning trace, and then continues token generation. Crucially, this system prompt provides minimal specific guidance about how to perform reasoning and query rewriting – this allows behaviors to be learned in RL training in a manner that (hopefully) is more flexible and general (Chu et al., 2025).

In training, the search agent performs rollouts of token generation and search. The final answer is extracted and we compute a very simple reward: 1 if the prediction matches any of the target answers, and 0 otherwise. Reward is applied to all LLM-generated tokens uniformly, except for the retrieved tokens that are masked out during gradient computation. More formally (as in Search-R1 (Jin et al., 2025)) we learn the weights for the policy LLM, π_θ, conditioned on a retrieval engine $\mathcal{R}$ using a QA dataset, $\mathcal{D}$:

$$\max_{\pi_\theta} \mathbb{E}_{x\sim\mathcal{D},y\sim\pi_\theta(\cdot|x;\mathcal{R})} \left[r_\phi(x,y)\right] \\ - \beta\mathbb{D}_{\text{KL}}\left[\pi_\theta(y \mid x;\mathcal{R}) \,||\, \pi_{\text{ref}}(y \mid x;\mathcal{R})\right]$$

In the first term, the LLM generates tokens, y from the question x, conditioned on a retriever: $y \sim \pi_\theta(\cdot\|R)$. The reward model, $r_\phi(x,y)$, extracts the answer from the sequence and compares

[2] PubMed abstracts originally sourced from National Library of Medicine

against ground truth. In the second term, the policy LLM, π_θ, is discouraged from diverging too far from a reference LLM π_θ, which is the LLM's initial state. We use Group Relative Policy Optimization (GRPO) to optimize the LLM based on the samples; further details in Appendix H.

4 Results

To demonstrate the utility of our dataset, corpus, and benchmarks, we train the LLM with reinforcement learning with verifiable rewards (RLVR). Our experiments show that RLVR training improves performance on scientific paper question-answering evaluations. We also provide further quantitative and qualitative analysis.

4.1 Experiment details

Baseline methods We build our dataset to facilitate training with RLVR, which supervises only the final answer and thus, promises stronger generalization compared to methods with heavy scaffolding or with reasoning SFT (Guo et al., 2025; Chu et al., 2025). To validate this strategy, we compare RLVR training to baseline LLM training approaches that impose few assumptions: direct LLM inference, chain-of-thought prompting (Wei et al., 2022; Kojima et al., 2022), retrieval augmented generation (RAG) (Lewis et al., 2020), and PaperQA2 with the same retriever as other methods (Skarlinski et al., 2024). For a fair comparison, we apply the same base LLM that was used in agent training.

Search-R1 RLVR Training We follow the Search-R1 training setup (Li et al., 2025a) as described in Section 3.4, and experiment with two retrievers: BM25 and e5. We use eight A100s (80GB) for training, using GRPO for 150 steps (runtime: ca. 30 hrs). We have batch size 512 and minibatch size 256 for two gradient updates per batch (full configuration is in the code). The training framework is *verl* (Sheng et al., 2024). The base LLMs are Qwen2.5 3B and 7B, and we experiment with both *base* and *instruct* (Team, 2024; Yang et al., 2024).

Evaluation We evaluate with the test set of PaperSearchQA, and the BioASQ-factoid benchmark (Krithara et al., 2023) version that we release (Section 3.2). The evaluation metric is the same as the RL training reward term: the prediction must exactly match one of the ground-truth answers, which are all synonyms (for example target answer 'APOC3' has synonyms 'apolipoprotein C-III', 'apoC-III', 'apoCIII', 'apolipoprotein C-III', 'apolipoprotein C3', among others). The matching function includes 'normalization': conversion to lower case, stripping leading and trailing whitespace, removing articles like 'a' and 'the'.

	PaperSearchQA	BioASQ
Qwen2.5-3b-Instruct		
Direct	16.7	15.8
CoT	20.3	16.5
RAG	32.0	30.0
Search-o1	30.8	29.4
PaperQA2	32.4	33.1
SearchR1	41.6	35.5
Qwen2.5-7b-Instruct		
Direct	27.5	24.9
CoT	29.7	23.4
RAG	36.5	29.7
Search-o1	36.5	31.5
PaperQA2	37.1	32.8
SearchR1	51.0	44.8

Table 1: Main results of baselines vs Search-R1 training (Jin et al., 2025) that uses RLVR. The metric is accuracy, where 'correct' is exact match of prediction to target (or a synonym for the target). PaperSearchQA is the test set of our dataset, while BioASQ is a human-created evaluation. The RAG and Search-R1 systems used BM25 retrieval, and we compare to e5 retriever in the text.

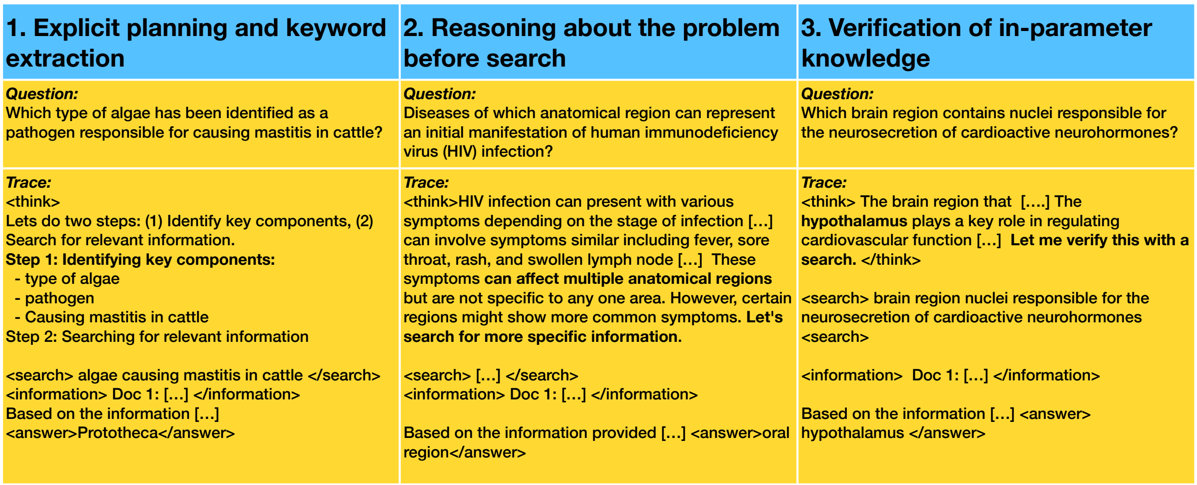
4.2 Quantitative Results

The main results are in Section 4.1, showing accuracy on the target benchmarks, and where the base model was *Qwen-instruct* (both, 3B and 7B variants). Training with RLVR (specifically using Search-R1 (Jin et al., 2025)) clearly leads to the strongest results. For the 3B LLMs, RL improves over RAG by 9.6 and 5.5 points for PaperSearchQA and BioASQ respectively. For 7B models, the difference is 14.5 and 9.3. RAG outperforms the retrieval-free methods by 17 points on average. Chain-of-thought prompting outperforms direct inference by only 1.2 points on average.

Table 2 shows per-category results for all models. The easiest overall categories are 'Biomarkers & diagnostics' and 'Protein function & signalling', while 'Genetic mutations' is the most challenging.

We perform further quantitative analysis and share these additional findings:

Semantic retrieval gives little benefit over syn-



1. Explicit planning and keyword extraction	2. Reasoning about the problem before search	3. Verification of in-parameter knowledge
Question: Which type of algae has been identified as a pathogen responsible for causing mastitis in cattle?	***Question:*** Diseases of which anatomical region can represent an initial manifestation of human immunodeficiency virus (HIV) infection?	***Question:*** Which brain region contains nuclei responsible for the neurosecretion of cardioactive neurohormones?
Trace: <think> Lets do two steps: (1) Identify key components, (2) Search for relevant information. **Step 1: Identifying key components:** - type of algae - pathogen - Causing mastitis in cattle Step 2: Searching for relevant information <search> algae causing mastitis in cattle </search> <information> Doc 1: […] </information> Based on the information […] <answer>Prototheca</answer>	***Trace:*** <think>HIV infection can present with various symptoms depending on the stage of infection […] can involve symptoms similar including fever, sore throat, rash, and swollen lymph node […] These symptoms **can affect multiple anatomical regions** but are not specific to any one area. However, certain regions might show more common symptoms. **Let's search for more specific information.** <search> […] </search> <information> Doc 1: […] </information> Based on the information provided […] <answer>oral region</answer>	***Trace:*** <think> The brain region that [….] The **hypothalamus** plays a key role in regulating cardiovascular function […] **Let me verify this with a search.** </think> <search> brain region nuclei responsible for the neurosecretion of cardioactive neurohormones <search> <information> Doc 1: […] </information> Based on the information […] <answer> hypothalamus </answer>

Figure 4: Three interesting behaviours that we observe in search agent traces. We bold some words for emphasis. Since traces are long, we abbreviate them, as indicated by '[...]'. These are discussed further in Section 4.3.

tactic retrieval For both RAG and RL training, we experimented with the BM25 syntactic retriever, and the e5 semantic retriever. While the semantic retriever should help search where exact keywords differ, the performance benefit was minor – within 2 points in all experiments. One possibility is that, even when paraphrasing questions, it must include certain technical keywords, which makes retrieval easier. Another possibility is that the e5 retriever under-performs for scientific domains (which involve highly technical terminology), thus removing the benefit of semantic retrieval.

LLMs encode scientific knowledge The retrieval-free baseline scores (from Section 4.1) are reasonably high, and scale with model size. For example on PaperSearchQA they score 20.3 and 29.7 for 3B and 7B models. This is probably explained by the fact that PubMed abstracts are easy to download, and so they likely appear in pretraining mixtures. Despite this data (probably) being seen by the model, memorization is far from perfect, so retrieval remains necessary.

Superior performance with model size is likely due to knowledge Averaged across benchmarks, Search-R1 outperforms CoT by 20.2 points for the 3B model and 21.4 for the 7B model. This suggests that the performance gain is due to improved parametric knowledge, and not due to superior capabilities in query formulation or comprehension.

Paraphrasing in data construction is beneficial In dataset construction, we observed that LLM-generated questions would often mirror keywords or phrasing from the source document in Section 3.1, and so we added a paraphrasing step to 50% of the QAs, allowing to compare non-paraphrases and paraphrased QAs. For SearchR1 trained on PaperSearchQA, non-paraphrased questions scored 57.2 while paraphrased questions scored 44.9, highlighting the importance of paraphrasing for sustaining question difficulty.

Training dynamics are similar to general-domain QA training environments The Search-R1 study (Jin et al., 2025) observed certain dynamics that we also observe. Specifically, we observed small performance difference between base and instruct models, albeit the base model required more training time to converge. We also found that training with GRPO was unstable, and reward would collapse to zero for some training runs – the base (non-instruct) models were generally more stable.

4.3 Qualitative Results

To better understand the system performance, we manually reviewed the reasoning traces for models at multiple stages in training. We highlight three prevalent patterns in Figure 4. The format of the traces includes reasoning inside '<think>' tokens and the final answer is in '<answer>' tokens. To perform retrieval, the LLM outputs text in '<search>' tags; the retrieved documents are dumped into the trace inside '<information>'.

Behavior 1 – explicit planning and keyword extraction. We find this pattern to be very common in later training. The model follows a clear and simple strategy common in RAG with rewriting: extracting the keywords for search and then combining them into a search query. After performing search, the LLM summarizes the final conclusion.

Behavior 2 – reasoning before search. Here, the LLM reasons about the question using only its parametric knowledge before performing any search. In the example problem, it observes that disease symptoms vary based on stage, and suggests symptoms from its own parametric knowledge. The trace acknowledges that it does not have the answer, and performs search. After viewing the retrieved information, the presence of earlier reasoning tokens may impact the final answer.

Behavior 3 – verification of in-parameter knowledge. The LLMs have sufficient knowledge to answer between 15% and 30% of questions (Section 4.1), so how does the agent behave when it already knows the answer? We find that it performs search anyway, but in the reasoning trace it will state its initial answer, and explicitly declare that it is doing further verification. Verification is generally good, since the LLM can gather more evidence for a reliable answer. More sophisticated systems however should only search when not confident in its initial answer.

Agent behavior becomes less varied with more training With more training, behavior 1 becomes much more common. We suspect this is due to lack of training data diversity – PaperSearchQAonly includes factoid-QA, and so this learned strategy is effective for most samples. Future systems trained on more QA types and elicit more varied behavior.

Very little reasoning after viewing documents After adding retrieved documents, the LLM tends to answer immediately, without explicit reasoning about document contents. This could be explained by comprehension being simpler with factoid-QA; it is also possible that RL training led to better comprehension due to parameter weight updates.

5 Discussion

We show that search agents can be trained using RL to perform question-answering by reasoning and gathering knowledge from scientific papers, a crucial intellectual part of science (Tsatsaronis et al., 2015; Hope et al., 2023). Search agents – and more generally RL-trained tool-use agents – are rapidly advancing in general-domain AI. Our aim in designing the training datasets, benchmarks, and corpus was to ensure compatibility with these methods. We hope that advances to general-domain agents – both in open research and in private labs – will translate to stronger capabilities in scientific literature understanding by leveraging our artifacts and others from the AI for science community.

While our datasets represent progress for scientific search agents, the scope is limited to only single-hop factoid-QA and simple retrieval over a database of abstracts – there is huge potential for further work. Interesting directions include factoid-QA designed to be multihop (Kim et al., 2025), answers with list-of-entities, and questions requiring extended answers or summaries (Krithara et al., 2023; Asai et al., 2024); these can require more complex agent planning behavior and fuzzy reward models. Even more ambitiously, future work could aim to resolve questions with conflicting evidence, like in critical literature review, (Lieberum et al., 2025; Polzak et al., 2025; Clark et al., 2025). Moreover, future datasets should consider that recent results in RLVR for (non-tool-use) LLMs are leveraging LLM-as-a-judge for reward modeling (Su et al., 2025; Gunjal et al., 2025). Meanwhile, other tool-use and search agent works consider text and images, which is relevant to scientific papers as well (Wu et al., 2025; Wang et al., 2025).

Other research directions are more specific to literature understanding applications. Agents could be equipped with tools and metadata that would be used by real scientists in their work, for example citation traversal and source reliability metrics. For example, one could implement a scoring on to what extent the conclusions extracted from a scientific article are supported by the figure images / data presented in the article – an assessment that is typically made by scientists when they deeply review literature. This could aid in valuing contradicting or diverging scientific results for a reply. Such metrics could be provided in the output, which could contain multiple answers with scores.

On a final note, our data generation pipeline is quite general – it could be adapted to generate QA datasets in other domains like chemistry, materials science, and computer science.

6 Conclusion

AI holds great potential to transform science. One exciting cluster of methods are LLM agents or multi-agent systems – sometimes called AI Scientists (Gao et al., 2024; Lu et al., 2024; Gottweis et al., 2025; Huang et al., 2025; Hope et al., 2023). This research program anticipates agents becoming more and more autonomous – first by performing well-defined tasks like data analysis and experimental execution (e.g., (Huang et al., 2025)) – and later performing more open-ended tasks (Hughes et al., 2024) like planning new experiments and even forming new hypotheses. But scientific fields are deeply knowledge-intensive: scientific discovery requires recalling, retrieving, and evaluating arcane information in the massive corpus of human knowledge. We therefore claim that future AI Scientist systems will require the capability of knowledge intensive search. Literature understanding is therefore fundamental to AI systems in science, and we believe that RL training of search agents – like in this paper – is an essential approach.

7 Limitations

First, the data generation pipeline is automatic and uses LLMs, which could lead to factually incorrect QAs. One source of risk is LLM hallucination, though the risk is small since each prompt has a smaller context, and we use strong LLMs (GPT-4.1).

Another risk from our data generation pipeline is that it is challenging to infer a 'general QA' from a single specific abstract. For example, an abstract might claim "mutation in gene X correlates with disease Y", and our pipeline might derive the question "what gene mutation is correlated with disease A?". But since we only have one abstract in context, we cannot be sure that 'gene mutation X' is the only answer – some other abstract might report that 'gene mutation Y' also correlates with the disease. In designing our data generation pipeline, expert review found such cases to be rare, and so we did not design complex mitigations. (It is possible that some such questions were avoided due to the parametric knowledge in the LLM generating the questions – GPT-4.1 – which is a more capable model than the smaller models used in these experiments). Future work that follow our data generation methodology could apply mitigations if needed. For example, if human review finds the issue prevalent for certain question categories, then that category could be excluded. Or, a workflow could be designed to retrieve all relevant papers to check for conflicts (which would be allowed a large retrieval budget).

In terms of scope, this is a first study in using RLVR to train search agents, so we restricted it to factoid QA. While this is a similar restriction to other early search agent papers, it represents only one of the possible question types important for real applications – we discuss future directions in Section 5. Likewise, our dataset covers scientific papers in biology & medicine, but not other domains commonly studied in AI for science like chemistry, materials science, computer science. However most AI for science papers have a similar limitation because significant domain expertise is required, making highly general studies challenging (Mirza et al., 2025; Burgess et al., 2025; Tang et al., 2025).

While the study provides resources towards building useful search agents for scientific practitioners, the derived agent system is a research prototype and is not suitable for real-world use. Apart from having a too-restricted scope, it has not undergone thorough evaluation needed for real-world deployment.

References

Josh Achiam, Steven Adler, Sandhini Agarwal, Lama Ahmad, Ilge Akkaya, Florencia Leoni Aleman, Diogo Almeida, Janko Altenschmidt, Sam Altman, Shyamal Anadkat, and 1 others. 2023. Gpt-4 technical report. *arXiv preprint arXiv:2303.08774*.

Anthropic. 2025. System card: Claude opus 4 & claude sonnet 4. Technical report, Anthropic. PDF, May 2025, "System card introduces Claude Opus 4 and Claude Sonnet 4".

Akari Asai, Jacqueline He, Rulin Shao, Weijia Shi, Amanpreet Singh, Joseph Chee Chang, Kyle Lo, Luca Soldaini, Sergey Feldman, Mike D'arcy, and 1 others. 2024. Openscholar: Synthesizing scientific literature with retrieval-augmented lms. *arXiv preprint arXiv:2411.14199*.

James Burgess, Jeffrey J Nirschl, Laura Bravo-Sánchez, Alejandro Lozano, Sanket Rajan Gupte, Jesus G Galaz-Montoya, Yuhui Zhang, Yuchang Su, Disha Bhowmik, Zachary Coman, and 1 others. 2025. Microvqa: A multimodal reasoning benchmark for microscopy-based scientific research. In *Proceedings of the Computer Vision and Pattern Recognition Conference*, pages 19552–19564.

Qiguang Chen, Libo Qin, Jinhao Liu, Dengyun Peng, Jiannan Guan, Peng Wang, Mengkang Hu, Yuhang Zhou, Te Gao, and Wanxiang Che. 2025. Towards reasoning era: A survey of long chain-of-thought for reasoning large language models. *arXiv preprint arXiv:2503.09567*.

Tianzhe Chu, Yuexiang Zhai, Jihan Yang, Shengbang Tong, Saining Xie, Dale Schuurmans, Quoc V Le, Sergey Levine, and Yi Ma. 2025. Sft memorizes, rl generalizes: A comparative study of foundation model post-training. *arXiv preprint arXiv:2501.17161*.

Justin Clark, Belinda Barton, Loai Albarqouni, Oyungerel Byambasuren, Tanisha Jowsey, Justin Keogh, Tian Liang, Christian Moro, Hayley O'Neill, and Mark Jones. 2025. Generative artificial intelligence use in evidence synthesis: A systematic review. *Research Synthesis Methods*, pages 1–19.

Julien Delile, Srayanta Mukherjee, Anton Van Pamel, and Leonid Zhukov. 2024. Graph-based retriever captures the long tail of biomedical knowledge. *arXiv preprint arXiv:2402.12352*.

Jiazhan Feng, Shijue Huang, Xingwei Qu, Ge Zhang, Yujia Qin, Baoquan Zhong, Chengquan Jiang, Jinxin Chi, and Wanjun Zhong. 2025. Retool: Reinforcement learning for strategic tool use in llms. *arXiv preprint arXiv:2504.11536*.

Adam R Ferguson, Jessica L Nielson, Melissa H Cragin, Anita E Bandrowski, and Maryann E Martone. 2014. Big data from small data: data-sharing in the'long tail'of neuroscience. *Nature neuroscience*, 17(11):1442–1447.

Shanghua Gao, Ada Fang, Yepeng Huang, Valentina Giunchiglia, Ayush Noori, Jonathan Richard Schwarz, Yasha Ektefaie, Jovana Kondic, and Marinka Zitnik. 2024. Empowering biomedical discovery with ai agents. *Cell*, 187(22):6125–6151.

Juraj Gottweis, Wei-Hung Weng, Alexander Daryin, Tao Tu, Anil Palepu, Petar Sirkovic, Artiom Myaskovsky, Felix Weissenberger, Keran Rong, Ryutaro Tanno, and 1 others. 2025. Towards an ai co-scientist. *arXiv preprint arXiv:2502.18864*.

Xinyan Guan, Jiali Zeng, Fandong Meng, Chunlei Xin, Yaojie Lu, Hongyu Lin, Xianpei Han, Le Sun, and Jie Zhou. 2025. Deeprag: Thinking to retrieve step by step for large language models. *arXiv preprint arXiv:2502.01142*.

Anisha Gunjal, Anthony Wang, Elaine Lau, Vaskar Nath, Bing Liu, and Sean Hendryx. 2025. Rubrics as rewards: Reinforcement learning beyond verifiable domains. *arXiv preprint arXiv:2507.17746*.

Daya Guo, Dejian Yang, Haowei Zhang, Junxiao Song, Ruoyu Zhang, Runxin Xu, Qihao Zhu, Shirong Ma, Peiyi Wang, Xiao Bi, and 1 others. 2025. Deepseek-r1: Incentivizing reasoning capability in llms via reinforcement learning. *arXiv preprint arXiv:2501.12948*.

Xanh Ho, Anh-Khoa Duong Nguyen, Saku Sugawara, and Akiko Aizawa. 2020. Constructing a multi-hop qa dataset for comprehensive evaluation of reasoning steps. *arXiv preprint arXiv:2011.01060*.

Tom Hope, Doug Downey, Daniel S Weld, Oren Etzioni, and Eric Horvitz. 2023. A computational inflection for scientific discovery. *Communications of the ACM*, 66(8):62–73.

Kexin Huang, Serena Zhang, Hanchen Wang, Yuanhao Qu, Yingzhou Lu, Yusuf Roohani, Ryan Li, Lin Qiu, Junze Zhang, Yin Di, and 1 others. 2025. Biomni: A general-purpose biomedical ai agent. *bioRxiv*, pages 2025–05.

Edward Hughes, Michael Dennis, Jack Parker-Holder, Feryal Behbahani, Aditi Mavalankar, Yuge Shi, Tom Schaul, and Tim Rocktaschel. 2024. Open-endedness is essential for artificial superhuman intelligence. *arXiv preprint arXiv:2406.04268*.

Aaron Jaech, Adam Kalai, Adam Lerer, Adam Richardson, Ahmed El-Kishky, Aiden Low, Alec Helyar, Aleksander Madry, Alex Beutel, Alex Carney, and 1 others. 2024. Openai o1 system card. *arXiv preprint arXiv:2412.16720*.

Bowen Jin, Hansi Zeng, Zhenrui Yue, Jinsung Yoon, Sercan Arik, Dong Wang, Hamed Zamani, and Jiawei Han. 2025. Search-r1: Training llms to reason and leverage search engines with reinforcement learning. *arXiv preprint arXiv:2503.09516*.

Qiao Jin, Bhuwan Dhingra, Zhengping Liu, William W Cohen, and Xinghua Lu. 2019. Pubmedqa: A dataset for biomedical research question answering. *arXiv preprint arXiv:1909.06146*.

Mandar Joshi, Eunsol Choi, Daniel S Weld, and Luke Zettlemoyer. 2017. Triviaqa: A large scale distantly supervised challenge dataset for reading comprehension. *arXiv preprint arXiv:1705.03551*.

Yunsoo Kim, Yusuf Abdulle, and Honghan Wu. 2025. Biohopr: A benchmark for multi-hop, multi-answer reasoning in biomedical domain. *arXiv preprint arXiv:2505.22240*.

Takeshi Kojima, Shixiang Shane Gu, Machel Reid, Yutaka Matsuo, and Yusuke Iwasawa. 2022. Large language models are zero-shot reasoners. *Advances in neural information processing systems*, 35:22199–22213.

Anastasia Krithara, Anastasios Nentidis, Konstantinos Bougiatiotis, and Georgios Paliouras. 2023. Bioasq-qa: A manually curated corpus for biomedical question answering. *Scientific Data*, 10(1):170.

Tom Kwiatkowski, Jennimaria Palomaki, Olivia Redfield, Michael Collins, Ankur Parikh, Chris Alberti, Danielle Epstein, Illia Polosukhin, Jacob Devlin, Kenton Lee, and 1 others. 2019. Natural questions: a benchmark for question answering research. *Transactions of the Association for Computational Linguistics*, 7:453–466.

Jakub Lála, Odhran O'Donoghue, Aleksandar Shtedritski, Sam Cox, Samuel G Rodriques, and Andrew D White. 2023. Paperqa: Retrieval-augmented generative agent for scientific research. *arXiv preprint arXiv:2312.07559*.

Nathan Lambert, Jacob Morrison, Valentina Pyatkin, Shengyi Huang, Hamish Ivison, Faeze Brahman, Lester James V Miranda, Alisa Liu, Nouha Dziri, Shane Lyu, and 1 others. 2024. T\" ulu 3: Pushing frontiers in open language model post-training. *arXiv preprint arXiv:2411.15124*.

Yoonjoo Lee, Kyungjae Lee, Sunghyun Park, Dasol Hwang, Jaehyeon Kim, Hong-in Lee, and Moontae Lee. 2023. Qasa: advanced question answering on scientific articles. In *International Conference on Machine Learning*, pages 19036–19052. PMLR.

Patrick Lewis, Ethan Perez, Aleksandra Piktus, Fabio Petroni, Vladimir Karpukhin, Naman Goyal, Heinrich Küttler, Mike Lewis, Wen-tau Yih, Tim Rocktäschel, and 1 others. 2020. Retrieval-augmented generation for knowledge-intensive nlp tasks. *Advances in neural information processing systems*, 33:9459–9474.

Xiaoxi Li, Guanting Dong, Jiajie Jin, Yuyao Zhang, Yujia Zhou, Yutao Zhu, Peitian Zhang, and Zhicheng Dou. 2025a. Search-o1: Agentic search-enhanced large reasoning models. *arXiv preprint arXiv:2501.05366*.

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

Judith-Lisa Lieberum, Markus Toews, Maria-Inti Metzendorf, Felix Heilmeyer, Waldemar Siemens, Christian Haverkamp, Daniel Böhringer, Joerg J Meerpohl, and Angelika Eisele-Metzger. 2025. Large language models for conducting systematic reviews: on the rise, but not yet ready for use—a scoping review. *Journal of Clinical Epidemiology*, 181:111746.

Chris Lu, Cong Lu, Robert Tjarko Lange, Jakob Foerster, Jeff Clune, and David Ha. 2024. The ai scientist: Towards fully automated open-ended scientific discovery. *arXiv preprint arXiv:2408.06292*.

Adrian Mirza, Nawaf Alampara, Sreekanth Kunchapu, Martiño Ríos-García, Benedict Emoekabu, Aswanth Krishnan, Tanya Gupta, Mara Schilling-Wilhelmi, Macjonathan Okereke, Anagha Aneesh, and 1 others. 2025. A framework for evaluating the chemical knowledge and reasoning abilities of large language models against the expertise of chemists. *Nature Chemistry*, pages 1–8.

Anastasios Nentidis, Georgios Katsimpras, Anastasia Krithara, Salvador Lima López, Eulália Farré-Maduell, Luis Gasco, Martin Krallinger, and Georgios Paliouras. 2023. Overview of bioasq 2023: The eleventh bioasq challenge on large-scale biomedical semantic indexing and question answering. In *International Conference of the Cross-Language Evaluation Forum for European Languages*, pages 227–250. Springer.

OpenAI. 2025a. Introducing deep research. https://openai.com/index/introducing-deep-research/. Accessed 2025-07-27.

OpenAI. 2025b. Introducing openai o3 and o4-mini. https://openai.com/index/introducing-o3-and-o4-mini/. Accessed 2025-07-27.

Krista Opsahl-Ong, Michael J Ryan, Josh Purtell, David Broman, Christopher Potts, Matei Zaharia, and Omar Khattab. 2024. Optimizing instructions and demonstrations for multi-stage language model programs. *arXiv preprint arXiv:2406.11695*.

Christopher Polzak, Alejandro Lozano, Min Woo Sun, James Burgess, Yuhui Zhang, Kevin Wu, and Serena Yeung-Levy. 2025. Can large language models match the conclusions of systematic reviews? *arXiv preprint arXiv:2505.22787*.

Cheng Qian, Emre Can Acikgoz, Qi He, Hongru Wang, Xiusi Chen, Dilek Hakkani-Tür, Gokhan Tur, and Heng Ji. 2025. Toolrl: Reward is all tool learning needs. *arXiv preprint arXiv:2504.13958*.

Stephen E Robertson and Steve Walker. 1994. Some simple effective approximations to the 2-poisson model for probabilistic weighted retrieval. In *SIGIR'94: Proceedings of the Seventeenth Annual International ACM-SIGIR Conference on Research and Development in Information Retrieval, organised by Dublin City University*, pages 232–241. Springer.

Timo Schick, Jane Dwivedi-Yu, Roberto Dessì, Roberta Raileanu, Maria Lomeli, Eric Hambro, Luke Zettlemoyer, Nicola Cancedda, and Thomas Scialom. 2023. Toolformer: Language models can teach themselves to use tools. *Advances in Neural Information Processing Systems*, 36:68539–68551.

Sander Schulhoff, Michael Ilie, Nishant Balepur, Konstantine Kahadze, Amanda Liu, Chenglei Si, Yinheng Li, Aayush Gupta, HyoJung Han, Sevien Schulhoff, and 1 others. 2024. The prompt report: a systematic survey of prompt engineering techniques. *arXiv preprint arXiv:2406.06608*.

Zhihong Shao, Peiyi Wang, Qihao Zhu, Runxin Xu, Junxiao Song, Xiao Bi, Haowei Zhang, Mingchuan Zhang, YK Li, Y Wu, and 1 others. 2024. Deepseekmath: Pushing the limits of mathematical reasoning in open language models. *arXiv preprint arXiv:2402.03300*.

Guangming Sheng, Chi Zhang, Zilingfeng Ye, Xibin Wu, Wang Zhang, Ru Zhang, Yanghua Peng, Haibin Lin, and Chuan Wu. 2024. Hybridflow: A flexible and efficient rlhf framework. *arXiv preprint arXiv: 2409.19256*.

Michael D Skarlinski, Sam Cox, Jon M Laurent, James D Braza, Michaela Hinks, Michael J Hammerling, Manvitha Ponnapati, Samuel G Rodriques, and Andrew D White. 2024. Language agents achieve superhuman synthesis of scientific knowledge. *arXiv preprint arXiv:2409.13740*.

Huatong Song, Jinhao Jiang, Yingqian Min, Jie Chen, Zhipeng Chen, Wayne Xin Zhao, Lei Fang, and Ji-Rong Wen. 2025. R1-searcher: Incentivizing the search capability in llms via reinforcement learning. *arXiv preprint arXiv:2503.05592*.

Yi Su, Dian Yu, Linfeng Song, Juntao Li, Haitao Mi, Zhaopeng Tu, Min Zhang, and Dong Yu. 2025. Crossing the reward bridge: Expanding rl with verifiable rewards across diverse domains. *arXiv preprint arXiv:2503.23829*.

Hao Sun, Zile Qiao, Jiayan Guo, Xuanbo Fan, Yingyan Hou, Yong Jiang, Pengjun Xie, Yan Zhang, Fei Huang, and Jingren Zhou. 2025. Zerosearch: Incentivize the search capability of llms without searching. *arXiv preprint arXiv:2505.04588*.

Yingheng Tang, Wenbin Xu, Jie Cao, Weilu Gao, Steve Farrell, Benjamin Erichson, Michael W Mahoney, Andy Nonaka, and Zhi Yao. 2025. Matterchat: A multi-modal llm for material science. *arXiv preprint arXiv:2502.13107*.

Qwen Team. 2024. Qwen2.5: A party of foundation models.

Harsh Trivedi, Niranjan Balasubramanian, Tushar Khot, and Ashish Sabharwal. 2022. Interleaving retrieval with chain-of-thought reasoning for knowledge-intensive multi-step questions. *arXiv preprint arXiv:2212.10509*.

George Tsatsaronis, Georgios Balikas, Prodromos Malakasiotis, Ioannis Partalas, Matthias Zschunke, Michael R Alvers, Dirk Weissenborn, Anastasia Krithara, Sergios Petridis, Dimitris Polychronopoulos, and 1 others. 2015. An overview of the bioasq large-scale biomedical semantic indexing and question answering competition. *BMC bioinformatics*, 16(1):138.

David Wadden, Shanchuan Lin, Kyle Lo, Lucy Lu Wang, Madeleine van Zuylen, Arman Cohan, and Hannaneh Hajishirzi. 2020. Fact or fiction: Verifying scientific claims. *arXiv preprint arXiv:2004.14974*.

Liang Wang, Nan Yang, Xiaolong Huang, Binxing Jiao, Linjun Yang, Daxin Jiang, Rangan Majumder, and Furu Wei. 2022. Text embeddings by weakly-supervised contrastive pre-training. *arXiv preprint arXiv:2212.03533*.

Qiuchen Wang, Ruixue Ding, Yu Zeng, Zehui Chen, Lin Chen, Shihang Wang, Pengjun Xie, Fei Huang, and Feng Zhao. 2025. Vrag-rl: Empower vision-perception-based rag for visually rich information understanding via iterative reasoning with reinforcement learning. *arXiv preprint arXiv:2505.22019*.

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

Jinming Wu, Zihao Deng, Wei Li, Yiding Liu, Bo You, Bo Li, Zejun Ma, and Ziwei Liu. 2025. Mmsearch-r1: Incentivizing lmms to search. *arXiv preprint arXiv:2506.20670*.

An Yang, Baosong Yang, Binyuan Hui, Bo Zheng, Bowen Yu, Chang Zhou, Chengpeng Li, Chengyuan Li, Dayiheng Liu, Fei Huang, Guanting Dong, Haoran Wei, Huan Lin, Jialong Tang, Jialin Wang, Jian Yang, Jianhong Tu, Jianwei Zhang, Jianxin Ma, and 40 others. 2024. Qwen2 technical report. *arXiv preprint arXiv:2407.10671*.

Zhilin Yang, Peng Qi, Saizheng Zhang, Yoshua Bengio, William W Cohen, Ruslan Salakhutdinov, and Christopher D Manning. 2018. Hotpotqa: A dataset for diverse, explainable multi-hop question answering. *arXiv preprint arXiv:1809.09600*.

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

Yuxiang Zheng, Dayuan Fu, Xiangkun Hu, Xiaojie Cai, Lyumanshan Ye, Pengrui Lu, and Pengfei Liu. 2025. Deepresearcher: Scaling deep research via reinforcement learning in real-world environments. *arXiv preprint arXiv:2504.03160*.

A Dataset and Code Availability

Accessing data We release all artifacts on the Huggingface Hub at https://huggingface.co/PaperSearchQA/datasets, under an anonymous account for submission.

Accessing Code For anonymous submission, code is submitted as a zip file to Openreview.

Licenses Our dataset, PaperSearchQA, is released under a fully open license CC-BY-4.0, permitting redistribution, remixing, and commercial use. The data is derived from PubMed abstracts that have no [3]. The search corpus and the BioASQ evaluation set are sourced from the BioASQ project (Krithara et al., 2023; Tsatsaronis et al., 2015), and inherit their CC-BY-2.5 license.

B Ethical considerations

This paper advances systems that answer scientific questions from literature, but this presents some risks:

- Agents may retrieve and amplify outdated, retracted, or flawed studies without quality assessment mechanisms.
- Papers retrieved by the agent may have some selection bias that is poorly understood, thus impacting papers seen by scientists.
- Hallucinations in LLM outputs and incorrect QA responses may harm scientific practice.
- Our dataset was generated in an automated pipeline, which may have introduced errors.

Future deployments should consider uncertainty quantification, and source quality indicators. More broadly, the scientific community must develop its own standards for the appropriate use of LLM tools that consider these risks.

C Statement on use of LLMs

LLMs were used at many points in the project. Other than what is discussed in the main paper, we had these use cases:

- In project conception: brainstorming ideas; giving feedback and criticism on project plans; searching related work; summarizing and answering questions about specific related work.
- In project execution: LLMs for code generation in the Cursor IDE.
- Paper writing: rephrasing individual sentences.

D Further explanation of reinforcement learning with verifiable rewards (RLVR)

RLVR (Lambert et al., 2024) is a post-training procedure in which a language model is optimized only from whether its *final* output can be automatically verified as correct. At a high level, the model proposes a solution to a task, a separate verifier evaluates that solution, and the model is updated to make successful solutions more likely in the future.

Single-turn RLVR. Much of the earliest RLVR work uses a single-turn setting, where the model answers in one shot without explicit tool calls or multiple interaction steps. Given a query x, the model samples a final answer $y \sim \pi_\theta(\cdot \mid x)$, such as a free-form solution to a math problem or a code snippet. A verifier V then returns a (typically scalar) reward

$$r = V(x, y),$$

for example by exact-match against a reference answer, a numerical tolerance check, or running unit tests on the generated code. In many RLVR setups, r is binary ($r \in \{0, 1\}$) to indicate pass/fail, but the formulation also allows graded or shaped rewards (e.g., partial credit or the proportion of tests passed).

In this setting, the RLVR objective is

$$J(\theta) = \mathbb{E}_{x \sim \mathcal{D},\, y \sim \pi_\theta(\cdot|x)}[\, r \,],$$

which says: sample questions x from a data distribution $\mathcal{D}$, sample answers y from the model, and maximize the expected reward returned by the verifier. This captures the basic "generate–verify–reinforce" loop used in early RLVR for math and code.

Multi-step RLVR with trajectories. For agents that call tools or take multiple reasoning steps, it is helpful to view RLVR in a more general trajectory form. Given a query x, the model interacts with its environment to produce a trajectory

$$\tau = (o_0, a_0, o_1, a_1, \ldots, o_T, y),$$

[3] https://www.nlm.nih.gov/databases/download.html

where o_t are observations (e.g., tool outputs or intermediate text), a_t are actions (e.g., tool calls or tokens), and y is the final answer returned to the user. The single-turn setting above is a special case where there are no intermediate observations or actions and τ consists only of the generated answer y.

A verifier V now maps (x, τ) or (x, y) to a scalar reward

$$r = V(x, \tau).$$

The verifier can use only the final answer (e.g., exact match or unit tests) or the whole interaction (e.g., whether a sequence of tool calls satisfies some constraints). Let $\pi_\theta(\tau \mid x)$ denote the model's policy over trajectories; RLVR then maximizes

$$J(\theta) = \mathbb{E}_{x \sim \mathcal{D},\, \tau \sim \pi_\theta(\cdot|x)}\big[\, r \,\big].$$

When r is binary, this reduces to maximizing the probability that the verifier accepts the trajectory, but the same objective accommodates more general reward shapes.

In practice, $J(\theta)$ is maximized using policy-gradient methods. In our experiments we use Group Relative Policy Optimization (GRPO; see appendix H), a variant that uses group-normalized advantages, clipping, and a KL penalty to a reference policy. For intuition, one can view these methods as refinements of the basic REINFORCE estimator

$$\nabla_\theta J(\theta) \approx \mathbb{E}\Big[(r - b) \sum_{t=0}^{T} \nabla_\theta \log \pi_\theta(a_t \mid h_t)\Big],$$

where b is a baseline that reduces variance. High-reward trajectories increase the log-probabilities of their actions, while low-reward trajectories decrease them.

Relation to SFT and RLHF. RLVR differs from supervised finetuning (SFT) and RLHF in two key ways. First, RLVR uses only verifiable success or failure of the *final* output as a learning signal; there are no human-written labels on intermediate steps and no preference scores over partial generations. Second, credit assignment is purely outcome-based: all intermediate reasoning, tool calls, and textual tokens are reinforced or discouraged according to the reward returned by the verifier. This makes RLVR particularly natural for tasks where correctness can be automatically judged but good intermediate supervision is expensive or unavailable.

E Data construction pipeline

We show the prompts here. The prompts are long, so for more readablity, refer to the code at `data_gen/generate_questions_from_abstracts.py`

Here is the main data generation prompt mapping an abstract to QAs.

```
BACKGROUND
You are a domain-expert biomedical NLP
    assistant.
You are helping me to create an
    open-domain QA dataset.
The downstream task will read a query
    and require an agent to search over
    Pubmed abstracts

--------
YOUR TASK
I will provide you with title and
    abstract of a Pubmed article.
Your task is to create 3 new
    question-answer pairs.

--------
TYPES OF QUESTIONS
The questions should be 'factoid based'.
The answer should be a simple entity.
It should not be ambiguous.
Don't be pretentious.

--------
IMPORTANT NOTES
The question-answer pair will be used
    to evaluation question-answering
    systems with retrieval. Ths means
    the target system does not know
    which paper the question was
    sourced from. So an inappropriate
    question would be "What technology
    is used in this study to ...". or
    "what type of treatment is assessed
    in this study?" (where the study
    name is not specified).
If the question contains acronyms that
    are not well known, then explain
    the acronym.

--------
EXAMPLE CATEGORIES
Below are sample categories with sample
    questions.

Category: 1 - Genetic inheritance &
    disease-linked mutations
question: What gene is mutated in
    Sickle Cell Anemia?
answer: HBB
question: Which ultraconserved element
    is associated with Embryonic Stem
    Cells (ESC) self-renewal?
answer: T-UCstem1
question: Is Huntington's disease
    caused by a dominate or recessive
    gene?
answer: dominant

Category: 2 - Therapeutics, indications
    & clinical evidence
```

```
question: What is the most effective
    drug for oxaliplatin-induced
    neuropathy?
answer: Duloxetine
question: Which cancer is the BCG
    vaccine used for?
answer: Non-muscle Invasive Bladder
    Cancer
question: How many injections of CLS-TA
    did the patients participating in
    the PEACHTREE trial receive?
answer: two

Category: 3 - Protein function,
    localization & signalling/enzymatic
    interactions
question: Which histone mark
    distinguishes active from inactive
    enhancers?
answer: H3K27ac
question: Which component of the
    Influenza A Virus affects mRNA
    transcription termination?
answer: NS1
question: Which is the main calcium
    binding protein of the sarcoplasmic
    reticulum?
answer: Calsequestrin

Category: 4 - Experimental &
    computational methods, resources &
    acronyms
question: Which algorithm has been
    proposed for efficient storage of
    WGS variant calls?
answer: SeqArray
question: What is an acceptable
    sequence coverage(depth) required
    for human whole-exome sequencing?
answer: 30x-60x

Category: 5 - Disease causation &
    pathogens
question: Which is the most common
    disease attributed to malfunction
    or absence of primary cilia?
answer: [’Polycystic kidney disease’,
    ’PKD’]
question: What organism causes scarlet
    fever also known as scarletina?
answer: [’Group A Streptococcus’,
    ’Streptococcus pyogenes’]
question: The pathogen Fusarium
    graminearum affects what type of
    plant species?
answer: cereal crops

Category: 6 - Biomarkers & diagnostic
    tests
question: Salivary Cortisol is a
    biomarker for what
    disease/syndrome/condition?
answer: stress
question: What is the gold standard for
    a diagnosis of narcolepsy?
answer: [’Sleep study’, ’overnight
    polysomnography’]

Category: 7 - Bioinformatics databases
    & curated resources
question: Which R/bioconductor package
    has been developed to aid in
    epigenomic analysis?
answer: DeepBlueR
question: Which database associates
    human noncoding SNPs with their
    three-dimensional interacting genes?
answer: 3DSNP
question: What is the RESID database?
question: Which is the literature-based
    database of phenotypes?
answer: PheneBank

Category: 8 - Clinical grading &
    diagnostic scales / classification
    systems
question: What can be predicted with
    the Wells criteria?
answer: pulmonary embolism
question: Symptoms of which disorder
    are evaluated with the Davidson
    Trauma Scale?
answer: [’post-traumatic stress
    disorder’, ’PTSD’]
question: Which value of nuchal
    translucency thickness is set as
    the threshold for high-risk for
    Down Syndrome?
answer: 3mm

Category: 9 - Anatomical / cellular
    structures & localisation
question: Where is corticosterone
    synthesized?
answer: Adrenal glands
question: Which is the chromosome area
    that the human gene coding for the
    dopamine transporter (DAT1) is
    located to?
answer: 5p15.3
question: Where is the respirasome
    located?
answer: inner mitochondrial membrane

Category: 10 - Psychology and
    behavioral health
Question: Which psychomotor domain
    showed a significant difference
    between institutionalized and
    non-institutionalized sheltered
    children and adolescents?
Answer: Body awareness
Question: What ethical principle
    justifies actions that have both
    good and harmful effects, as long
    as the harm is not intended but
    only foreseen?
Answer: Rule of Double Effect
Questions: What psychological process
    during an incubation period is
    associated with enhanced creative
    problem solving?
Answer: Mind-wandering

--------

OUTPUT FORMAT
A single QA has tags
    ‘<question>...</question>‘, answer
    inside ‘<answer>...</answer>‘.
```

```
If the QA corresponds to one of the
    above categories put its number in
    <cat_num>...</cat_num> and category
    description in <cat>...</cat>.
Each QA should exist in its own tag
    <qa>...</qa>

Therefore the first 2 questions would
    be:
<qas>
   <qa> <question> ... </question>
      <answer> ... </answer>
      <cat_num> ... </cat_num>
      <cat> ... </cat>
   </qa>
   <qa>
        .....
   </qa>
   ...
</qas>

--------
TITLE AND ABSTRACT
{title_abstract}
"""
```

And here is the prompt for generating 'golden answers' or synonyms to the ground truth answer.

```
You are given a question that was
    written using a particular document
    as its main source. Your task is to
    rewrite the question so that it
    retains the original meaning and
    would result in the same correct
    answer, but uses different wording
    and phrasing. Important constraints:
Do not broaden or narrow the scope of
    the question.
Do not introduce ambiguity or alter
    clinical/technical context.
Make sure the correct answer remains
    exactly the same.
Your goal is to change the surface
    wording so that simple bag-of-words
    search (like BM25) may not easily
    match the original document, while
    an expert human or strong language
    model could still answer correctly.
Avoid copying any significant phrase
    (three or more words in sequence)
    from the original question.

Example:
- Original: What congenital abnormality
    can cause unilateral hydrocephalus
    in the perinatal period?
- Edited: Which birth defect present
    during the perinatal stage may
    result in hydrocephalus affecting
    only one side of the brain?

Output should be in tags like
    <question> ... </question>

Question: {question}
Answer: {answer}
```

F System prompt for Search-R1 LLM training

The LLM system prompt provides basic guidance about what tools are available, as well as guidance about putting the final answer in tags.

```
Answer the given question. You must
    conduct reasoning inside <think>
    and </think> first every time you
    get new information. After
    reasoning, if you find you lack
    some knowledge, you can call a
    search engine by <search> query
    </search> and it will return the
    top searched results between
    <information> and </information>.
    You can search as many times as
    your want. If you find no further
    external knowledge needed, you can
    directly provide the answer inside
    <answer> and </answer>, without
    detailed illustrations. For
    example, <answer> Beijing
    </answer>. Question: {question}\n
```

For baseline experiments we apply the same formatting instruction.

G Results: per-category performance

Since PaperQA2 has per-category labels (fig. 2), we report the main results split by these category values. The main results are in table 2.

	Data portion	3b models					7b models				
		Direct	CoT	RAG	PaperQA2	Search-R1	Direct	CoT	RAG	PaperQA2	Search-R1
Genetic mutations	3.6	12	17	40	20	27	18	18	45	19	26
Therapeutics & clinical evidence	17	17	23	31	28	38	27	32	37	32	46
Protein function & signalling	12.36	15	20	39	32	44	28	29	46	37	53
Methods & resources	26.36	14	16	26	25	35	26	25	30	27	37
Disease causation & pathogens	12.96	24	27	38	33	39	34	38	43	36	52
Biomarkers & diagnostics	10.38	20	19	26	34	46	29	32	30	40	56
Bioinformatics databases	0.16	13	25	13	100	100	13	13	13	100	100
Clinical scales & classifications	2.82	16	16	26	25	34	23	26	28	34	50
Anatomy & cellular localisation	8.74	13	22	39	24	32	27	30	42	28	37
Psychology & behavioural health	3.4	16	19	28	26	33	27	31	31	30	39

Table 2: Main results of baselines vs Search-R1 training (Jin et al., 2025) that uses RLVR. Unlike the table in the main results, we show the per-category scores, where the categories are defined in fig. 2.

H Training RLVR details

This section is single-column due to the large equation below. Continuing the description of the RL training algorithm from section 3.4, we leverage Group Relative Policy Optimization (GRPO) (Shao et al., 2024; Guo et al., 2025). At each iteration, we have the current policy π_θ, which we now temporarily call the ‘old policy’ π_{old}. For each question, x, GRPO computes multiple rollouts $\{y_1, y_2, \ldots, y_G\}$ using π_{old}, and we can now consider some averaging of rewards ina group. The policy model is then optimized by maximizing:

$$\mathcal{J}_{GRPO}(\theta) = \mathbb{E}_{x \sim \mathcal{D}, \{y_i\}_{i=1}^{G} \sim \pi_{\text{old}}(\cdot|x;\mathcal{R})} \left[\frac{1}{G} \sum_{i=1}^{G} \frac{1}{\sum_{t=1}^{|y_i|} I(y_{i,t})} \sum_{t=1:I(y_{i,t})=1}^{|y_i|} \min \left(\frac{\pi_\theta(y_{i,t}|x, y_{i,<t}; \mathcal{R})}{\pi_{\text{old}}(y_{i,t}|x, y_{i,<t}; \mathcal{R})} \hat{A}_{i,t}, \right. \right.$$

$$\left. \left. \text{clip}\left(\frac{\pi_\theta(y_{i,t}|x, y_{i,<t}; \mathcal{R})}{\pi_{\text{old}}(y_{i,t}|x, y_{i,<t}; \mathcal{R})}, 1-\epsilon, 1+\epsilon \right) \hat{A}_{i,t} \right) - \beta \mathbb{D}_{KL} \left[\pi_\theta || \pi_{\text{ref}} \right] \right], \tag{1}$$

Here, $\mathcal{R}$ is the retriever (as before), ϵ controls clipping range, and β controls KL penalty. We compute ‘advantages’ (rather than raw reward), $\hat{A}_{i,t}$ by normalizing rewards within each group of G responses by using group mean as baseline and group standard deviation for scaling.

The full training scripts with all hyperparameters are available in the released code.