

Scientific evidence and tool use

Tool interfaces and evidence records.

Reasoning agents, feedback, and scientific evidence

An answer can require information the model lacks

Imagine a generated research answer cites a measurement whose units and calibration status are absent. Fluent continuation cannot supply the missing record reliably.

The gap concerns external evidence. A more elaborate explanation of the same available text may leave the question unresolved.

This suggests giving the system a way to acquire the specific record or execute the required check, rather than only generate another answer.

Turning an evidence gap into a tool request

Turn the uncertainty into a request: retrieve the measurement with units, fetch its calibration record, or run a calculation on admissible data.

A response can reveal a new problem. Mixed units require conversion, while an expired reference requires different evidence rather than another numeric computation.

A tool interface therefore needs inputs, returned evidence and recoverable error states. The reasoning policy decides what response warrants the next action.

Choosing a source of external evidence

Candidate	Reason to choose it	Cost or limitation
Retrieve a source	The needed fact exists in an accessible record	A retrieved passage may not support the claim
Execute a calculation	Inputs and a valid procedure are available	Correct execution cannot validate bad inputs
Acquire new evidence	Existing records cannot settle the question	Measurements or expert attention impose costs

Choose the operation that supplies the missing evidence. Preserve provenance and distinguish a tool failure from a scientific result.

Pedagogical reconstruction

[Yao et al. \(2022\)](#)

[Schick et al. \(2023\)](#)

A schema checks form, not scientific validity

A valid JSON object can contain a wrong identifier, an unsupported claim, or a result from the wrong dataset version.

Validate units, ranges, identifier mappings, provenance, and relevant scientific assumptions in addition to syntax. Test how the agent behaves when a tool reports uncertainty or partial failure.

A reliable interface communicates these distinctions instead of compressing every result into a single success flag.

Retrieval can support claims when attribution is precise

ALCE (Automatic LLMs' Citation Evaluation; Gao et al., 2023) tests whether cited evidence supports generated statements. Link each passage to its document and location, and preserve relevant context.

A citation can be real while the claim is unsupported. Multiple citations can repeat the same underlying source.

Distinguish direct evidence, secondary discussion, and the agent's own inference so readers can trace the argument.

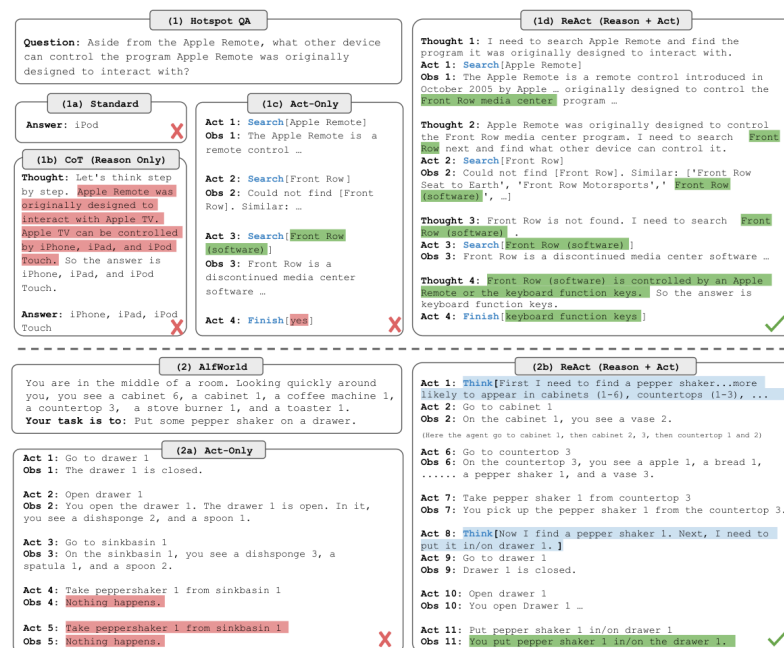
ReAct: the method

ReAct interleaves language-model reasoning, actions, and observations. A reasoning step can revise a plan; an action queries a source or changes an environment; the returned observation constrains the next step.

The paper evaluates question answering, fact verification, and interactive tasks.

The trace exposes a useful control interface, although a readable explanation is not necessarily a faithful account of the model's internal computation.

ReAct: a thought–action–observation trace



The trace separates model-generated thoughts and actions from returned environmental observations. Observations can add external information, whose reliability depends on the source or tool. A generated thought can reinterpret evidence without supplying a new observation.

ReAct: mechanism detail

(1d) ReAct (Reason + Act)

Thought 1: I need to search Apple Remote and find the program it was originally designed to interact with.

Act 1: `Search[Apple Remote]`

Obs 1: The Apple Remote is a remote control introduced in October 2005 by Apple ... originally designed to control the `Front Row media center` program ...

Thought 2: Apple Remote was originally designed to control the Front Row media center program. I need to search `Front Row` next and find what other device can control it.

Act 2: `Search[Front Row]`

Obs 2: Could not find [Front Row]. Similar: ['Front Row Seat to Earth', 'Front Row Motorsports', '`Front Row (software)`', ...]

Thought 3: Front Row is not found. I need to search `Front Row (software)` .

Act 3: `Search[Front Row (software)]`

Obs 3: Front Row is a discontinued media center software ...

Thought 4: `Front Row (software)` is controlled by an Apple Remote or the keyboard function keys. So the answer is keyboard function keys.

Act 4: `Finish[keyboard function keys]`



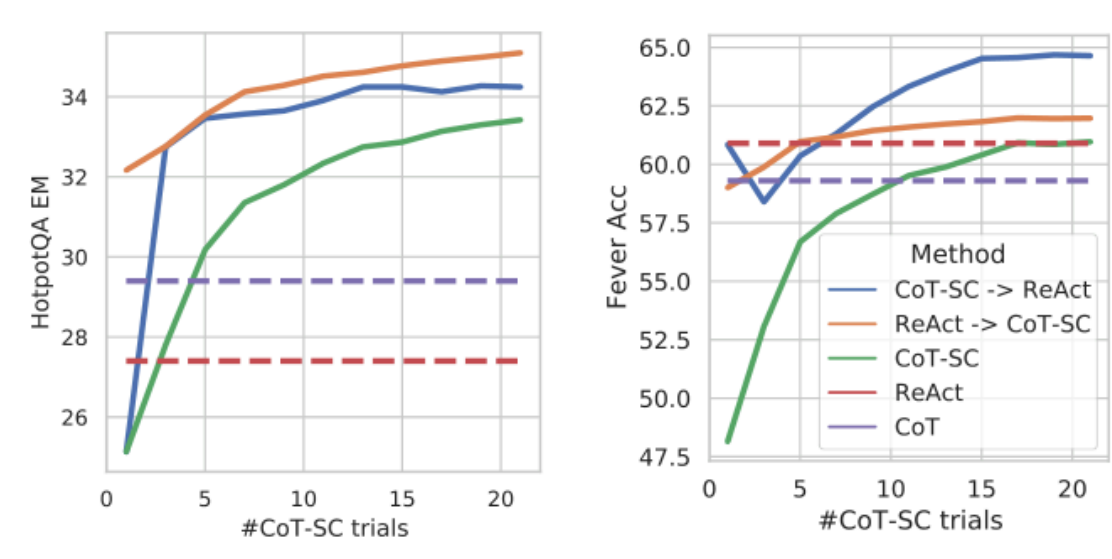
An observation can impose a new constraint on the next action. A responsive policy uses that constraint rather than simply continuing the original plan.

Evidence: tool use has a task-dependent effect

Prompt Method ^a	HotpotQA (EM)	Fever (Acc)
Standard	28.7	57.1
CoT (Wei et al., 2022)	29.4	56.3
CoT-SC (Wang et al., 2022a)	33.4	60.4
Act	25.7	58.9
ReAct	27.4	60.9
CoT-SC → ReAct	34.2	64.6
ReAct → CoT-SC	35.1	62.0
Supervised SoTA^b	67.5	89.5

With PaLM-540B, ReAct reports 27.4 HotpotQA exact match and 60.9 FEVER accuracy; CoT reports 29.4 and 56.3. Hybrid rows change the policy again. Metrics and tasks differ, so retrieval's grounding benefits coexist with possible search, interpretation and orchestration failures.

ReAct: interpretation and limits



The second visual reports comparisons under specific task and prompting conditions. Observable actions make auditing possible without guaranteeing correct interpretation of an observation.

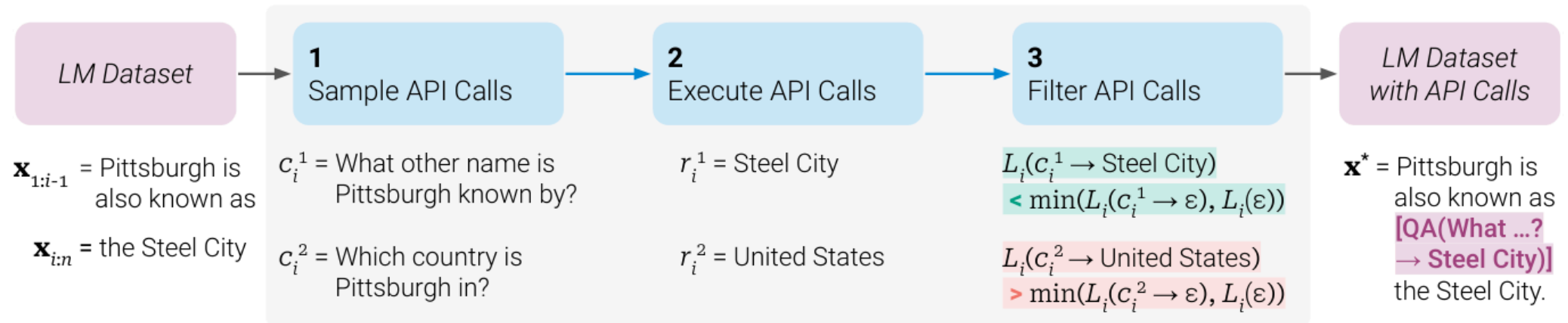
Toolformer: the method

Toolformer generates candidate API calls, executes them, and retains calls whose results improve a language-modelling loss criterion. The model is then trained on the augmented text.

This teaches decisions about when to call a tool, what arguments to supply, and how to use its response.

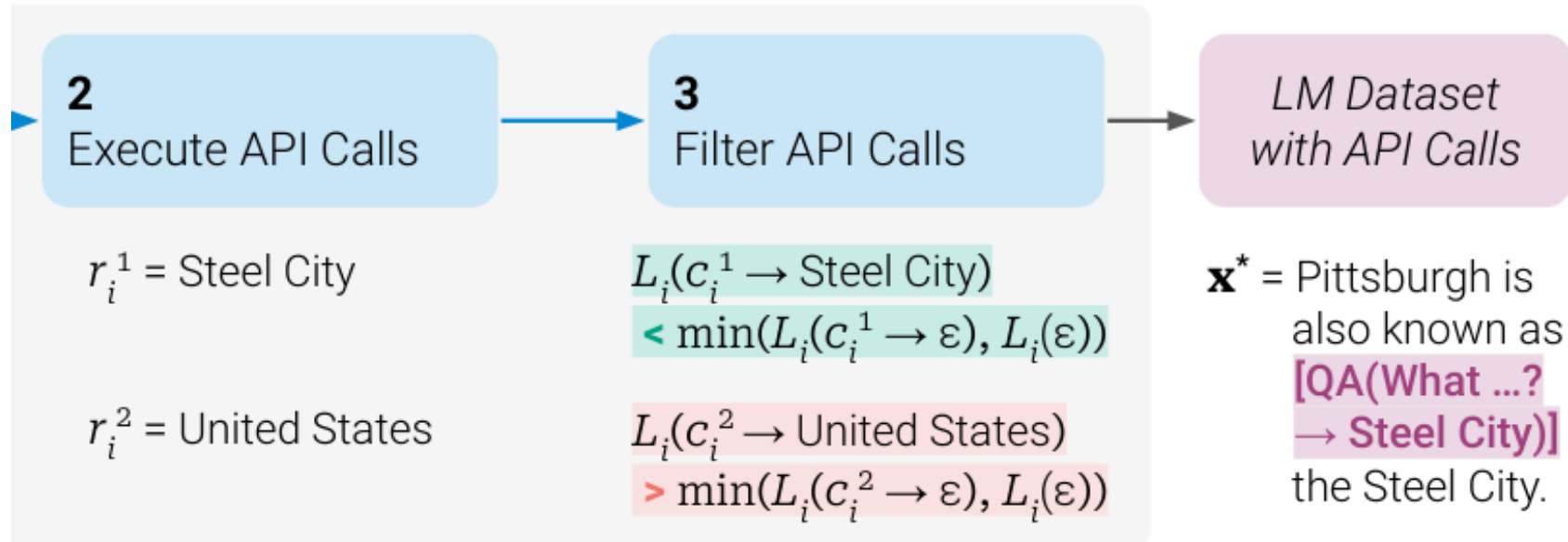
The selection signal rewards usefulness for predicting subsequent text; it does not directly measure scientific validity or end-to-end task utility.

Toolformer: API-call training pipeline



Candidate calls pass through execution and loss-based filtering before entering augmented training data. The loss comparison supplies the supervision signal for retaining a call.

Toolformer: mechanism detail



The filtering stage compares loss with access to a tool result against the paper's baseline condition. Executing candidate calls contributes to the training cost of this selection process.

Evaluation scope

The source evaluates tool-assisted performance on tasks suited to the included APIs, such as arithmetic and factual lookup.

The displayed examples illustrate call syntax and how results enter the continuation; examples are not themselves a benchmark aggregate.

Separate the capability of the external tool from the learned policy for invoking and interpreting it.

Toolformer: interpretation and limits

The New England Journal of Medicine is a registered trademark of [QA("Who is the publisher of The New England Journal of Medicine?") → Massachusetts Medical Society] the MMS.

Out of 1400 participants, 400 (or [Calculator(400 / 1400) → 0.29] 29%) passed the test.

The name derives from "la tortuga", the Spanish word for [MT("tortuga") → turtle] turtle.

The Brown Act is California's law [WikiSearch("Brown Act") → The Ralph M. Brown Act is an act of the California State Legislature that guarantees the public's right to attend and participate in meetings of local legislative bodies.] that requires legislative bodies, like city councils, to hold their meetings open to the public.

The training distribution, API definitions, and filtering objective constrain transfer. A useful calculator call is narrow evidence for an invocation policy, not a demonstration of reliable use of arbitrary scientific software.

Execution needs reproducible inputs and environments

Record the code or command, input identifiers and hashes, relevant software versions, parameters, random seeds, and output artefacts.

Reproducing a numerical result does not validate the assumptions that generated it, but it permits diagnosis and independent checking.

When a tool produces nondeterministic output, report the variation and preserve enough context to reproduce the evaluation distribution rather than promising identical bytes.

Treat tool output as data with a defined authority

External documents and retrieved text can contain irrelevant instructions or malformed content. A tool contract should separate data fields from instructions that control the agent.

Authorise actions based on the workflow and declared task, not on commands embedded inside a retrieved paper or webpage.

Evaluate this boundary with benign adversarial examples, such as a document that asks the agent to cite an unrelated result.

Local tool reliability and end-to-end reliability

A tool can have excellent isolated accuracy while a multi-step pipeline fails through identifier mismatches, unit errors, or unsupported inference between stages.

Measure call validity, result interpretation, claim support, and end-to-end task quality. Compare against a simpler tool pipeline using the same data.

The final trace should make it possible to identify whether the error came from the tool, orchestration, or scientific reasoning.

DSPy: the method

DSPy expresses language-model pipelines through declarative modules and compiles their parameters, demonstrations, or prompts against an explicit metric.

This separates program structure from particular prompt strings and makes optimisation systematic.

The metric, training examples, and evaluation split become central parts of the software specification; compilation can exploit a flawed objective as effectively as manual tuning.

DSPy: task-specific pipeline comparisons

Program	Compilation	Training	GPT-3.5		Llama2-13b-chat	
			Dev	Test	Dev	Test
vanilla	none	n/a	24.0	25.2	7.0	9.4
	fewshot	trainset	33.1	–	4.3	–
	bootstrap	trainset	44.0	–	28.0	–
	bootstrap×2	trainset	64.7	61.7	37.3	36.5
	+ensemble	trainset	62.7	61.9	39.0	34.6
CoT	none	n/a	50.0	–	26.7	–
	fewshot	trainset	63.0	–	27.3	–
	fewshot	+human_CoT	78.6	72.4	34.3	33.7
	bootstrap	trainset	80.3	72.9	43.3	–
	+ensemble	trainset	88.3	81.6	43.7	–
reflection	none	n/a	65.0	–	36.7	–
	fewshot	trainset	71.7	–	36.3	–
	bootstrap	trainset	83.0	76.0	44.3	40.2
	+ensemble	trainset	86.7	–	49.0	46.9

The table compares compiled pipelines and models on specified tasks. Baselines, metrics, retrieval access and demonstrations define the scope of the comparison.

DSPy optimizes a composed program

Program component	First attempt	What optimization can change
First-hop retrieval	Retrieve the entity named in the question	Instruction and demonstrations
Second-hop query	Repeat the original wording	Use the first result to target the missing relation
Answer generation	Combine retrieved passages	Instruction, examples and evidence handling
Development metric	Score the whole program's output	Select settings that work across cases

DSPy keeps the program structure explicit while optimizing language-model components. Development improvement still requires held-out checking.

Evaluation scope

The study examines retrieval and reasoning pipelines using multiple models and optimisation configurations.

The second table supplies another task-specific comparison; absolute values must be interpreted using its stated metric and evaluation split.

These results support optimisation of modular LM programs under the tested conditions, not a guarantee that compilation yields a valid scientific workflow.

DSPy: interpretation and limits

Program	Compiler	GPT-3.5				Llama2-13b-chat			
		Dev		Test		Dev		Test	
		Ans	Psg	Ans	Psg	Ans	Psg	Ans	Psg
vanilla	fewshot	34.3	n/a	31.5	n/a	27.5	n/a	21.8	n/a
CoT_RAG	fewshot	36.4	36.0	29.8	34.4	34.5	36.0	28.0	34.4
	bootstrap	42.3	36.0	–	–	38.3	36.0	32.9	34.4
react	none	20.3	–	–	–	20.0	–	–	–
	+human_r	33.0	–	–	–	28.3	–	–	–
	bootstrap	31.0	–	–	–	24.7	–	–	–
	bootstrap×2	39.0	–	–	–	40.0	–	–	–
multihop	fewshot	36.9	38.3	31.2	40.8	34.7	32.0	31.3	30.8
	bootstrap	48.7	47.0	39.6	43.8	42.0	48.3	36.4	43.5
	ensemble	54.7	–	45.6*	–	50.0	–	41.0	–

Tuning and selecting a pipeline repeatedly on the same evaluation set can overfit it. Separate development feedback from final evaluation and record compiler settings and data lineage.

A research answer is a chain of claims

Retrieval supplies documents or records. Extraction identifies relevant statements. Synthesis connects them to the question. Each transition can lose a qualifier or introduce an unsupported inference.

Evaluate the chain at the level of individual claims, with source versions and the scope of support retained.

Three evidence checks

Check	Unit evaluated	Typical failure
Retrieval relevance	Retrieved item	Relevant evidence missed
Citation support	Claim-source pair	Citation does not entail claim
Claim coverage	Answer claim	Important claim uncited

A high result on one check can coexist with a low result on another.

Human intervention changes the regime

Record whether an expert supplies a search query, corrects an extracted value, chooses a hypothesis or approves a final conclusion. These actions contribute different capabilities.

Report assisted and unassisted performance separately, together with the amount of expert effort. A final approval does not reveal which intermediate claims the expert actually checked.

A claim-level evidence audit

Claim	Citation attached	Source supports it?
A	Yes	Yes
B	Yes	Yes
C	Yes	No
D	No	Not established

Synthetic claim audit inspired by ALCE's support/coverage distinction. With one citation per cited claim, support is 2/3 and supported-claim coverage is 2/4; this is not a reported ALCE result.

Evidence acquisition can be a budgeted set problem

Operation	Calculation or state	Consequence
Claim weights	A:3, B:1, C:1, D:4	Value of established support
Actions	q1:{A,B}, cost 2; q2:{B,C}, cost 1	Overlapping support
Third action	q3:{D}, cost 2	Budget is 3
Feasible pairs	q1+q2: value 5; q2+q3: value 6	Choose q2+q3

Synthetic deterministic acquisition model. A covered claim here means verified support under the stipulated action, rather than a document merely being retrieved.

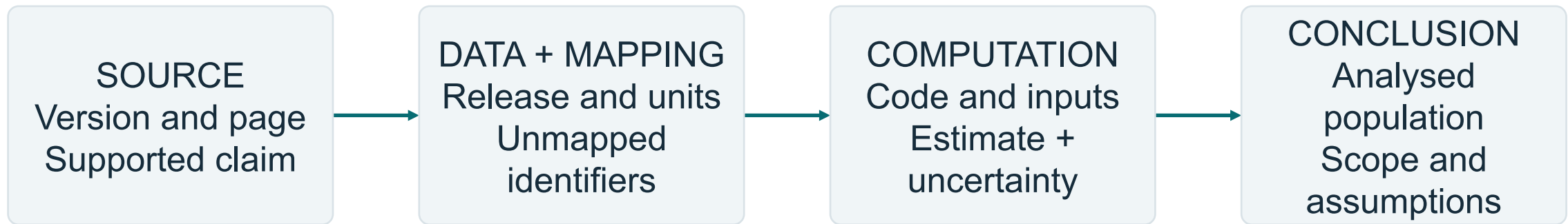
Worked evidence chain: a public-data analysis

An agent retrieves a paper, downloads a public table, computes a group difference, and writes a conclusion.

The paper uses gene symbols from an older annotation release; the downloaded table uses current identifiers. A mapping step is required before the analysis is meaningful.

This synthetic workflow contains a common provenance hazard: each tool can return a valid object even when the objects do not refer to the same scientific entities.

Specify the evidence record



Every arrow needs a valid transformation or support relation. A successful call can conceal mismatched identifiers or units. Retain checksums, mapping coverage, and excluded records so the final claim can be traced to its actual inputs. A precise estimate cannot repair an unreported change in the analysed population.

A successful call can still break the argument

Suppose the identifier mapper silently drops 15% of rows. The analysis script completes and produces a statistically precise group difference.

The precision describes the retained data under the analysis model. It does not address bias introduced by non-random missing mappings.

A meaningful contract should return mapping coverage and dropped identifiers. The agent must then assess whether the missingness changes the population or comparison represented by the result.

AgentIR separates input from retriever training

Retriever condition	Reasoning included?	BrowseComp-Plus accuracy
Base embedding model	No	48.67%
Change the retrieval input only	Yes	55.54%
Retrain on synthetic research queries	No	59.40%
AgentIR: both changes	Yes	66.27%

Embeds query + reasoning trace. March 2026 preprint, v1 Table 2: Tongyi-DeepResearch, no Visit tool. The $\approx 68\%$ headline uses Visit.

A useful tool call changes expected decision value

$$\text{VOI}(e) = \mathbb{E}_y \left[\max_a \mathbb{E}[U(a, H) | D, y, e] \right] - \max_a \mathbb{E}[U(a, H) | D]$$

Value of information (VOI) compares expected utility after a tool result with the best current decision. H is the uncertain scientific state and a a possible final decision.

Subtract tool cost before choosing whether to call it. The result model must represent uncertainty or bias in that tool.

A syntactically valid call or a lower language-model loss does not by itself establish positive value for the scientific decision.

An interface test should expose semantic mismatch

Unmapped, duplicated and version-changed identifiers can alter a pipeline's exclusions and denominator. Explicit handling of these cases connects data processing to the scope of its scientific claim.

Verify the downstream analysis against an independently constructed reference on this controlled input.

The small example isolates interface semantics. Evaluation on realistic data addresses whether the checks improve the complete workflow.

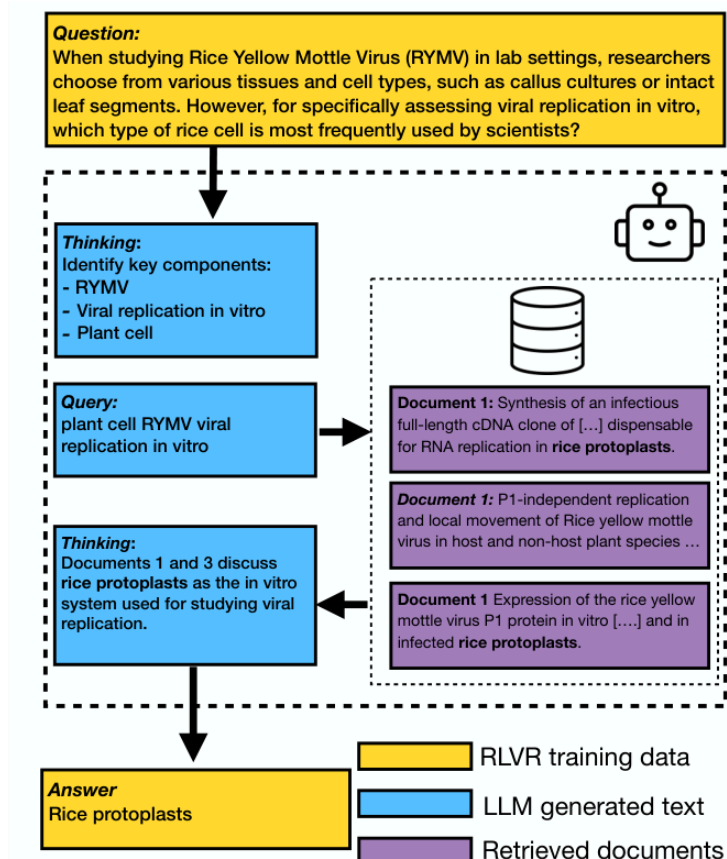
Recent research: PaperSearchQA

PaperSearchQA trains agents to search a large corpus of biomedical abstracts and answer technical factoid questions using verifiable rewards.

The source provides a corpus, question data, and evaluations of retrieval-and-reasoning policies.

The literature snapshot is dated 11 September 2026. Findings remain conditional on the source version and evaluation conditions. Recent preprints contribute reported results with distinct validation requirements.

PaperSearchQA: search and answer workflow



The architecture connects a question, search actions, retrieved evidence and a final answer. The reward-bearing output and any separate check of evidential support define what the training signal measures.

Recent methods in context

Recursive Language Models (preprint; revised 2026). Treats long context as an external environment that can be inspected programmatically and processed through recursive model calls. Nested calls can increase cost unpredictably; long-context benchmark gains do not establish universal task efficiency.

ResearchBench (ACL Findings 2026). Separates inspiration retrieval, hypothesis composition and ranking in literature-derived scientific tasks. Retrospective reconstruction and a publication-date filter do not establish prospective novelty or eliminate contamination.

FrontierScience (OpenAI report, 2025/26). Separates Olympiad questions from expert-written scientific research sub-tasks with rubric-based grading. Bounded scientific tasks cover only part of discovery, and scores depend on task selection and the evaluator.

References (1/2)

[Shunyu Yao, Jeffrey Zhao, Dian Yu et al. \(2022\). ReAct: Synergizing Reasoning and Acting in Language Models](#)

[Timo Schick, Jane Dwivedi-Yu, Roberto Dessì et al. \(2023\). Toolformer: Language Models Can Teach Themselves to Use Tools](#)

[Omar Khattab, Arnav Singhvi, Paridhi Maheshwari et al. \(2023\). DSPy: Compiling Declarative Language Model Calls into Self-Improving Pipelines](#)

[Kexin Huang, Serena Zhang, Hanchen Wang et al. \(2025\). Biomni: A General-Purpose Biomedical AI Agent](#)

[Kathryn Chaloner, Isabella Verdinelli \(1995\). Bayesian Experimental Design: A Review](#)

[Zijian Chen, Xueguang Ma, Shengyao Zhuang et al. \(2026\). AgentIR: Reasoning-Aware Retrival for Deep Research Agents](#)

References (2/2)

James Burgess, Jan N. Hansen, Duo Peng et al. (2026). PaperSearchQA: Learning to Search and Reason over Scientific Papers with RLVR

Alex L. Zhang, Tim Kraska, Omar Khattab (2025). Recursive Language Models

Yujie Liu, Zonglin Yang, Tong Xie et al. (2025). ResearchBench: Benchmarking LLMs in Scientific Discovery via Inspiration-Based Task Decomposition

Miles Wang, Robi Lin, Kat Hu et al. (2025). FrontierScience: Evaluating AI's Ability to Perform Expert-Level Scientific Tasks

Tianyu Gao, Howard Yen, Jiatong Yu et al. (2023). Enabling Large Language Models to Generate Text with Citations