

AI Agents for Science

Lecture 10, October 29: Benchmarking and Evaluation

Instructor: Ian Foster

TA: Alok Kamatar



Crescat scientia; vita excolatur

CMSC 35370 -- <https://agents4science.github.io>
<https://canvas.uchicago.edu/courses/67079>

Curriculum

1) Why AI agents for science?

AI agents and the sense-plan-act-learn loop. Scientific Discovery Platforms (SDPs): AI-native systems that connect reasoning models with scientific resources.

2) Frontiers of Language Models

Surveys frontier reasoning models: general-purpose LLMs (GPT, Claude), domain-specific foundation models (materials, bio, weather), and hybrids. Covers techniques for eliciting better reasoning: prompting, chain-of-thought, retrieval-augmented generation (RAG), fine-tuning, and tool-augmented reasoning.

3) Systems for Agents

Discusses architectures and frameworks for building multi-agent systems, with emphasis on inter-agent communication, orchestration, and lifecycle management.

4) Retrieval Augmented Generation (RAG) and Vector Databases

Covers how to augment reasoning models with external knowledge bases, vector search, and hybrid retrieval methods.

Curriculum

5) Tool Calling

Introduces methods for invoking external tools from reasoning models. Focus on model context protocol (MCP), schema design, and execution management.

6) HPC Systems and Self Driving Labs

How SDPs connect to HPC workflows and experimental labs. Covers distributed coordination, robotics, and federated agents.

7) Human–AI Workflows

Explores how scientists and agents collaborate: trust boundaries, interaction design, and debugging.

8) Benchmarking and Evaluation

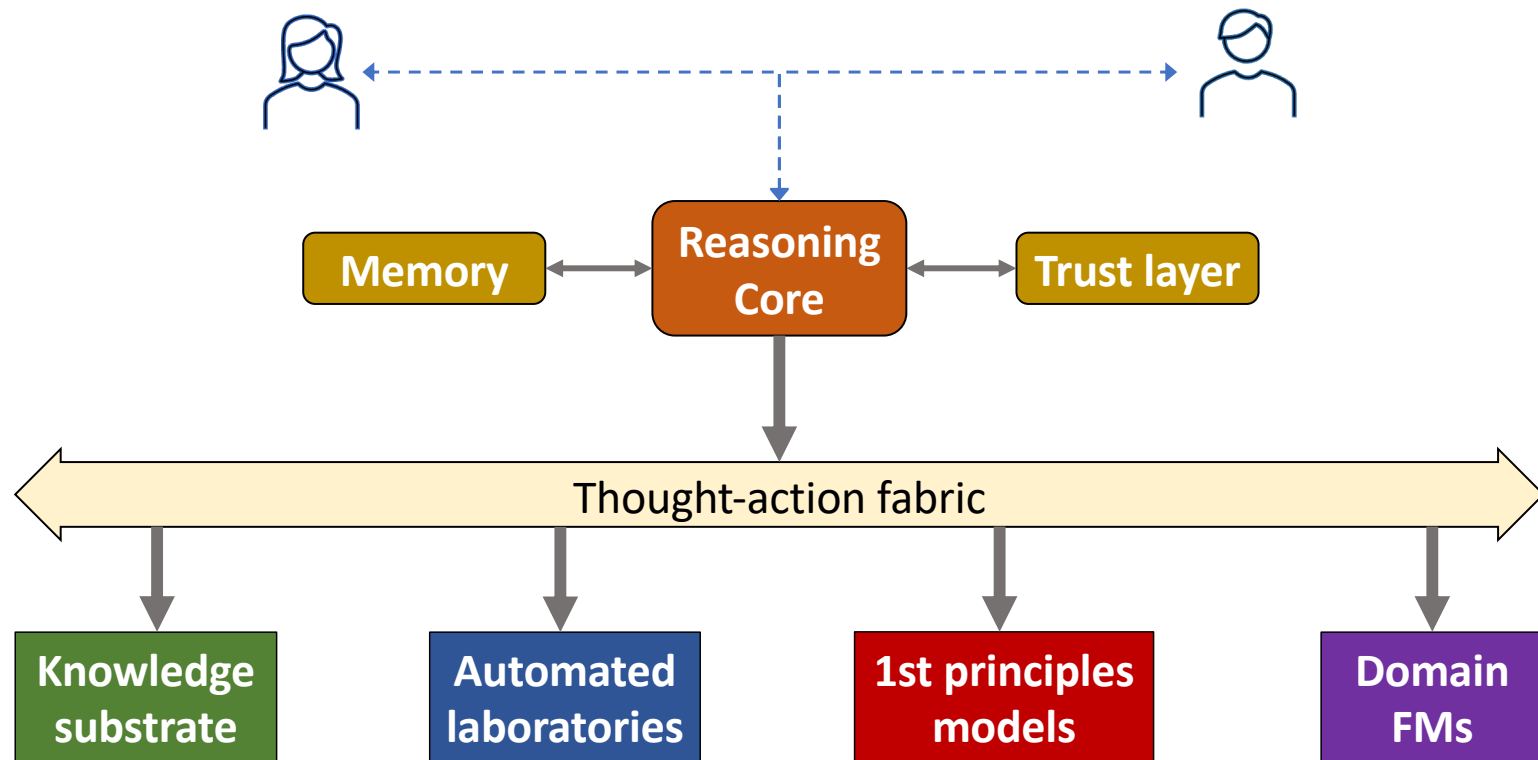
Frameworks for assessing agents and SDPs: robustness, validity, and relevance.

Readings

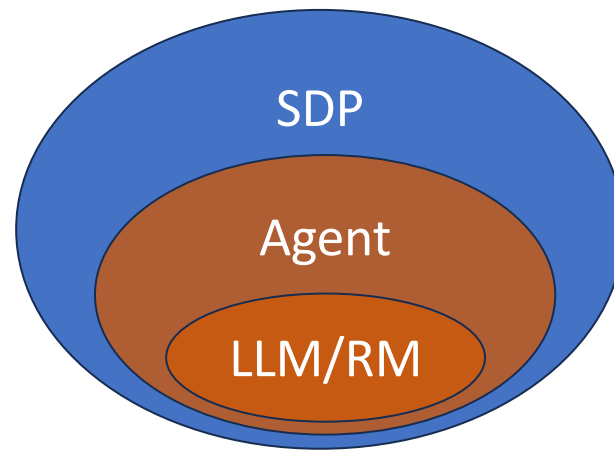
- [MLE-bench: Evaluating Machine Learning Agents on Machine Learning Engineering](#)
- Holistic Agent Leaderboard
- [EAIRA: Establishing a Methodology for Evaluating AI Models as Scientific Research Assistants](#)

Question: How would you evaluate an AI agent or multi-agent system designed for use in science?

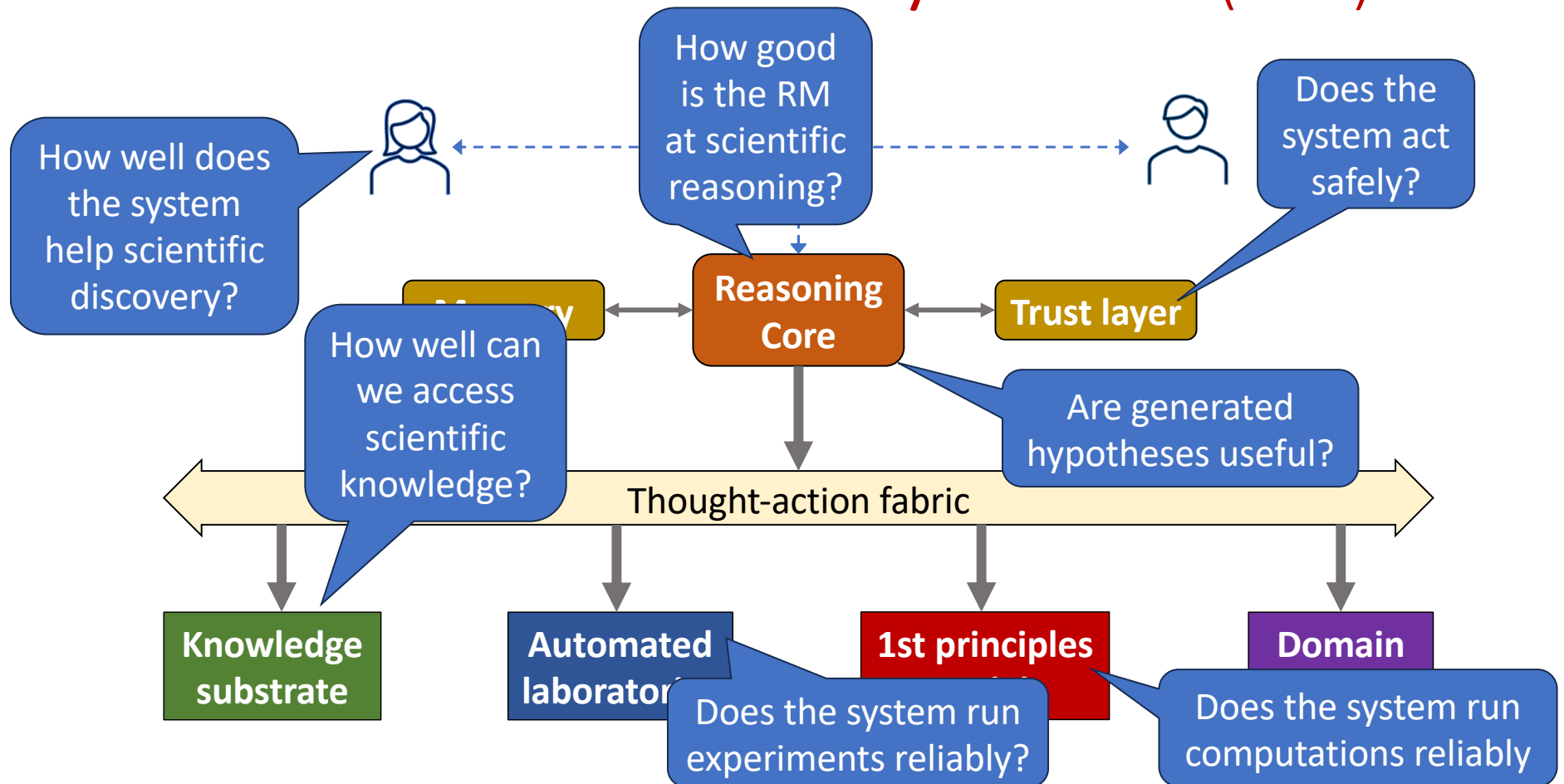
Recall the Scientific Discovery Platform (SDP)



“Reasoning core” encompasses LLM/RM & agent



Recall the Scientific Discovery Platform (SDP)



Dimension	Typical Questions
Task Success	Did the agent complete the assigned experiment or simulation correctly?
Scientific Validity	Are the generated hypotheses, analyses, or data products scientifically sound?
Efficiency	How many iterations, compute cycles, or lab actions per useful result?
Autonomy and Coordination	How well does the agent plan, delegate, and adapt without human input?
Robustness and Safety	Does it recover from errors, detect anomalies, or propagate uncertainty?
Human Alignment	Are human goals, ethics, and constraints respected?

Measure the workflow, not just the model

- **Task success:** Scientific objective reached? Yes/no + quality
- **Human effort:** Interaction count, minutes of attention
- **Intervention/override rate** (and “regret”): Times the human undid an agent action
- **Calibration:** Brier score on agent confidence vs. outcomes
- **Safety incidents / near-misses**
- **Latency & cost:** Queue time, tokens, compute \$
- **Reproducibility:** Can we replay to the same artifact?
- **Knowledge carry-over:** Does the system perform better on similar tasks over time?

Levels of evaluation

Level	Evaluation Focus	Example Metrics
Scientific Discovery Platform Evaluation	End-to-end scientific outcomes	Reproducibility, discovery rate, human–AI collaboration
Agent Evaluation	Behavior and performance in context	Task success, efficiency, robustness, safety
Reasoning Model Evaluation	Quality of inference and logic	Faithfulness, process quality, step consistency
LLM / Model Evaluation	Core model capabilities	Accuracy, F1, BLEU, log-likelihood, perplexity

Evaluation of ML models: 10 key ideas

1. **Purpose:** Evaluation determines how well a model generalizes beyond its training data and whether its predictions are reliable, useful, and safe in real-world contexts
2. **Types:** Models are typically assessed through *quantitative metrics* (e.g., accuracy, F1, BLEU, RMSE) and *qualitative or human-judged criteria* (e.g., coherence, reasoning quality, faithfulness)
3. **Data Splits and Generalization:** Using disjoint training, validation, and test sets—or cross-validation—is essential to measure true generalization rather than memorization or overfitting

Evaluation of ML models: 10 key ideas

4. **Baselines and Ablations:** Comparisons to strong baselines and systematic ablation studies reveal whether model improvements stem from genuine advances or confounding factors like data size or prompt tuning
5. **Robustness and Uncertainty:** Good evaluations test performance under distribution shift, noise, and adversarial conditions, and quantify uncertainty through confidence scores or ensembles
6. **Fairness and Bias:** Assessments should check whether model behavior varies unfairly across demographic, domain, or temporal subgroups

Evaluation of ML models: 10 key ideas

- 7. Efficiency and Cost:** Evaluation increasingly includes compute, memory, latency, and energy costs, reflecting sustainability and practical deployability
- 8. Reproducibility:** Publishing code, datasets, seeds, and evaluation scripts is crucial to verify results and ensure others can replicate and extend the findings
- 9. Human-in-the-Loop and Task Utility:** Especially for generative and agentic systems, evaluation should include *end-to-end task success* and *human productivity gains*, not just intrinsic metrics
- 10. Continuous Evaluation:** In dynamic settings (e.g., online agents or evolving data), evaluation must be an ongoing process, integrating monitoring and feedback to detect performance drift

Challenges of measurement

- “If you can't measure it, you can't improve it”
 - “I often say that when you can measure what you are speaking about, and express it in numbers, you know something about it; but when you cannot measure it, when you cannot express it in numbers, your knowledge is of a meagre and unsatisfactory kind; it may be the beginning of knowledge, but you have scarcely, in your thoughts, advanced to the stage of science, whatever the matter may be.” – Lord Kelvin
- “When a measure becomes a target, it ceases to be a good measure”
 - Goodhart’s Law

Evaluating LLMs

An LLM is an **open-ended generative system that can** produce diverse outputs; thus, when evaluating we need to look at more than “accuracy”: also meaning, reasoning, and usefulness

Intrinsic / static benchmarks

Measure the model's *raw capabilities* via fixed datasets and metrics

- **Knowledge & reasoning:** MMLU, ARC, GSM8K, HellaSwag, TruthfulQA
- **Code & math:** HumanEval, MBPP, GSM8K, MATH, SciCode
- **Language understanding:** GLUE, SuperGLUE

Metrics: Accuracy, exact match, BLEU, ROUGE, F1, log-likelihood

E.g., MMLU (Massive Multitask Language Understanding)

- Covers 57 subjects across humanities, STEM, social sciences, etc.
- E.g., from “high school physics”:
 - **Question:** The plates of a capacitor are charged to a potential difference of 5 V. If the capacitance is 2 mF, what is the charge on the positive plate?
 - **Options:** ["0.005 C", "0.01 C", "0.02 C", "0.5 C"]
 - **Answer:** B
- E.g., from “college physics”:
 - **Question:** The coefficient of static friction between a small coin and the surface of a turntable is 0.30. The turntable rotates at 33.3 revolutions per minute. What is the maximum distance from the center of the turntable at which the coin will not slide?
 - **Options:** ["0.024 m", "0.048 m", "0.121 m", "0.242 m"]
 - **Answer:** D

GSM8K: Grade School Math 8K

Benchmark for evaluating how well language models perform multi-step mathematical reasoning in natural language

Question: John has 5 apples. He buys 3 more packs of 4 apples each.
How many apples does he have now?

Reasoning: 3 packs \times 4 apples = 12 apples
 $5 + 12 = 17$ apples

Answer: 17

E.g., SuperGLUE
(GLUE = General Language Understanding Evaluation)

- **Passage:** Barq's is an American soft drink. Its brand of root beer is notable for having caffeine. Barq's, created by Edward Barq and bottled since the turn of the 20th century, is owned by the Barq family but bottled by the Coca-Cola Company. It was known as Barq's Famous Olde Tyme Root Beer until 2012.
- **Question:** is barq's root beer a pepsi product
- **Answer:** No

Metrics

- Accuracy: Fraction of predictions that are exactly correct
- Exact match: 1 if two strings are identical, 0 otherwise
- BLEU (Bilingual Evaluation Understudy): n-gram overlap
- ROUGE (Recall-Oriented Understudy for Gisting Evaluation): text recall
- F1: Combine precision and recall
- Log-likelihood: Likelihood of text given reference distributions

BLEU (Bilingual Evaluation Understudy)

BLEU measures how much **n-gram overlap** there is between a **candidate translation** and one or more **reference translations**, weighted toward *precision* (what fraction of candidate's n-grams appear in the reference)

$$\text{BLEU} = BP \cdot \exp\left(\sum_{n=1}^N w_n \log p_n\right)$$

E.g.:

- **Reference:** the cat is on the mat
- **Candidate:** the cat sat on the mat
- Unigrams capture word choice
- Higher-order n-grams capture correct word order and grammar
- No semantics

$$p1=5/6=0.833$$

$$p2=3/5=0.6$$

$$p3=1/4=0.25$$

$$p4=0/3=0$$

n	Candidate n-grams	#	# in ref
1-grams	the, cat, sat, on, the, mat	6	5
2-grams	the cat, cat sat, sat on, on the, the mat	5	3
3-grams	the cat sat, cat sat on, sat on the, on the mat	4	1
4-grams	the cat sat on, cat sat on the, sat on the mat	3	0

Evaluation of reasoning models

A **reasoning model** is an AI system trained or tuned to produce and evaluate *chains of inference*, not just fluent text. It typically:

- **Represents problems** in a form suitable for inference: symbolic, probabilistic, or linguistic
- **Applies multi-step reasoning**, decomposing a complex problem into smaller steps
- **Maintains intermediate state**: thoughts, hypotheses, subgoals
- **Checks or revises its reasoning**: self-consistency, self-verification
- **Produces interpretable traces** of its reasoning process
- **Outperforms** shallow LLMs on benchmarks like GSM8K, BBH, GPQA, MATH

Leading closed-source reasoning models

Model	Org	Year	Key Traits	What's New
OpenAI o1 (and o1-mini)	OpenAI	2025	Performs <i>internal long reasoning passes</i> before responding; trains on process supervision (PRM800K)	Hidden “slow thinking” phase before emitting answers; excels on math and science reasoning
GPT-4-Turbo (Reasoning mode)	OpenAI	2024	Supports chain-of-thought (CoT) and self-consistency internally	Improved planning and reasoning via longer context and reflection
Anthropic Claude 3.5 Sonnet / Opus	Anthropic	2024	High-quality natural-language reasoning and critique loops	Trained with constitutional feedback to reason safely and logically
DeepSeek-R1 / R1-Zero	DeepSeek	2024–25	Fully open research model with <i>multi-round self-improvement reasoning</i>	Reasoning distillation via reinforcement learning on process correctness
Google Gemini 2.0 Reasoning	Google DeepMind	2024	Multimodal reasoning, step-by-step inference across text, vision, and code	Integrates tool use, retrieval, and reflection; strong on science tasks

Leading open-source reasoning models

Model	Developer	Key Method	Notes
Mistral-R	Mistral AI	Instruction + process supervision	Open-weights model emphasizing reasoning traces
Qwen-2.5-Math / Qwen-Reasoner	Alibaba	Reinforcement learning from step-level feedback	Strong on GSM8K, MATH, and symbolic tasks
Llama-3.1-Reasoning (experimental)	Meta	Chain-of-thought-augmented fine-tuning	Internal reasoning datasets + PRM fine-tuning
Yi-Lightning-Reasoning	01.AI	Multi-step reflective reasoning training	Efficient reasoning variant tuned for CoT stability
WizardMath / Orca2 / Phi-3-Reasoning	Microsoft Research	Distillation from reasoning traces	Trained on synthetic CoT datasets for math and logic

Specialized scientific reasoning models

Model	Domain	Key Idea
AlphaGeometry 2 (DeepMind)	Geometry reasoning	Combines LLM reasoning with formal theorem checking
LeanDojo / ProofNet Models	Formal math	Generate and verify proofs in Lean / Isabelle
SciReason / SciCode / Tau-Bench Agents	Scientific workflows	Multimodal reasoning for experiment planning and result interpretation
ChemCrow	Chemistry reasoning	Tool-augmented reasoning using domain APIs
CoScientist / MADSci Agents	Autonomous scientific discovery	Combine reasoning models with control and data agents

A fairly recent benchmark performance snapshot

Model	GSM8K (EM)	MATH	GPQA	ARC-C	Comment
o1	~95 %	~85 %	~75 %	~90 %	Deep internal reasoning phase
DeepSeek-R1	~93 %	~82 %	~72 %	~88 %	Strong open alternative
GPT-4-Turbo	~92 %	~80 %	~70 %	~87 %	Baseline high-end reasoning
Claude 3.5 Opus	~90 %	~78 %	~70 %	~85 %	Coherent chain-of-thought
Mistral-R	~88 %	~70 %	—	—	Open-weights reasoning model

EM = Exact Match




Levels of reasoning model of evaluation

- **Outcome-only:** Check the final answer
 - E.g., GSM8K, MATH, BBH, ARC-Challenge, GPQA
- **Process-level:** Evaluate each step in a reasoning chain
 - E.g., PRM800K, COT-Eval, FOLIO, EntailmentBank
- **Faithfulness:** Verify that reasoning chain leads to final answer
 - E.g., counterfactual testing (change premise), self-critique validation
- **Robustness:** Stability across phrasing, noise, sampling
 - E.g., paraphrase consistency, addition of distractors
- **Generalization and transfer:** Ability to combine skills on unseen tasks
 - E.g., BBH, MATH, AR-Math, LogicNLI, ProofWriter
- **Human- and reward-model-based:** Use another strong model, or human expert, to evaluate




PRM800K: Process Reward Model 800 K




- A step-level human-labeled dataset for evaluating and supervising the process of reasoning rather than only the final answer
- 800K step-level labels across 75K solutions to 12K problems

The denominator of a fraction is 7 less than 3 times the numerator. If the fraction is equivalent to $\frac{2}{5}$, what is the numerator of the fraction? (Answer:)

   Let's call the numerator x .

   So the denominator is $3x-7$.

   We know that $x/(3x-7) = 2/5$.

   So $5x = 2(3x-7)$.

   $5x = 6x - 14$.

   So $x = 7$.

<https://arxiv.org/pdf/2305.20050>

EAIRA: Multi-faceted evaluation methodology

Proposed Methodology				
Techniques	MCQ Benchmarks	Open Response Benchmarks	Lab Style Experiments	Field Style Experiments
Main Goal	Testing knowledge breadth, basic reasoning	Testing knowledge depth, planning, reasoning	Realistic testing	Realistic trend analysis and weakness diagnosis
Problem Type	Predetermined , Fixed Q&As with known solutions	Predetermined , Fixed Free-Response Problems with known solutions	Individual Human Defined Problems with unknown solutions	Many Human Defined Problems with (un)known solutions
Verification	Automatic response verification	Automatic or Human response verification	Humans detailed response analysis	Scalable automatic summary of human response
Examples	Astro, Climate, AI4S (multi-domain), Existing Benchmarks	SciCode, ALDbench	see "lab style experiments"	see "field style experiments"
Cross Cutting Aspects	← Trust and Safety (ChemRisk), Uncertainty Quantification, Scalable Software Infrastructure (STAR) →			

4 complementary evaluation techniques to comprehensively assess the capabilities of LLMs as scientific assistants.

(Prior work by others, **Prior work by authors**, **New work**)

EAIRA: A Methodology for Evaluating AI Models as Scientific Research Assistants, <https://arxiv.org/pdf/2502.20309>.

Skills evaluated by each evaluation technique. Lab-style experiments focus on detailed analysis in controlled environments. Field-style experiments focus on analyzing researcher–LLM interactions at scale in natural settings.

Tasks in the Scientific Research Workflow	Scientific Research Assistant Expected Skills	Techniques			
		MCQ Benchmarks	Open Response Benchmarks	Lab Style Experiments (Focus on human detailed analysis in controlled environment)	Field Style Experiments (Focus on automated scalable analysis in natural environment)
Research question/Problem formulation	Problem understanding, Knowledge	X	X	X	X
Conduct initial preliminary experiments/data collection	Experiment Design/ Data retrieval			X	X
Literature search	Knowledge, Retrieval, Literature Understanding			X	X
Hypothesis generation	Propose relevant research directions, existing/analytical solutions	X	X	X	X
Hypothesis testing	Propose pertinent experiments, observations, simulations		X	X	X
Test results analysis	Propose/Use relevant data analysis techniques	X		X	X
Report writing	Generate a research report describing the scientific problem, the related literature, the proposed solution, its evaluation and the lessons learned and conclusions)			X	X

<https://arxiv.org/pdf/2502.20309>




AstroMLab comprises 4,425 multiple-choice questions curated from the Annual Review of Astronomy and Astrophysics, covering a broad range of astrophysical topics

Paper ID: 2023ARA&A..61..131F

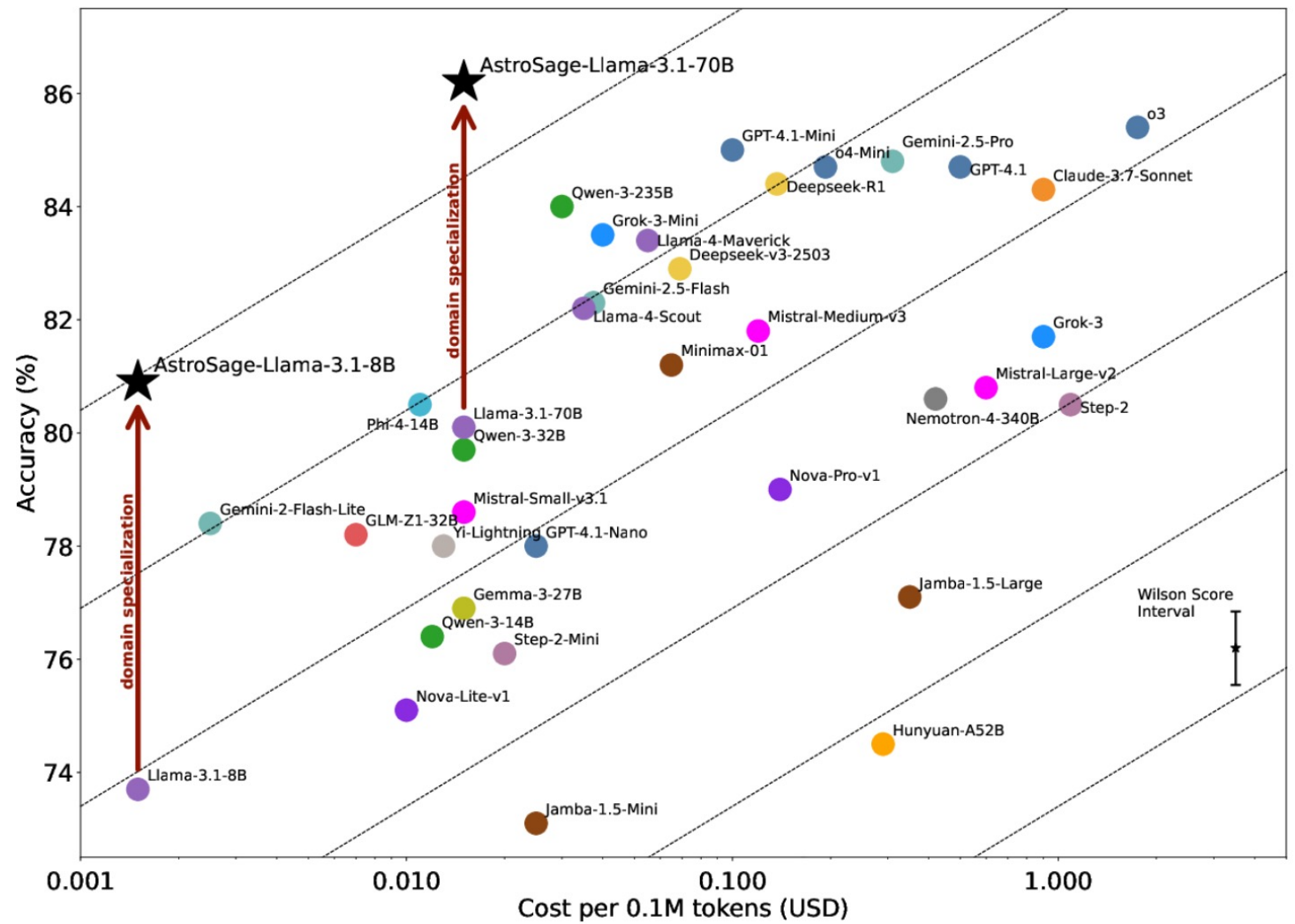
Question: The properties of the circumgalactic medium (CGM) primarily depend on the competition between:
(A) Star formation rate and supernova feedback.
(B) Gas cooling and stellar winds.
(C) Gravity-driven infall and gas cooling.
(D) Magnetic fields and thermal conduction.

Correct Answer: C

Explanation: The article explicitly states that the defining characteristic of the CGM is the balance between gravity pulling gas inwards and cooling processes that allow gas to lose pressure and condense. This balance dictates whether the CGM is predominantly hot (slow cooling) or cold (rapid cooling).

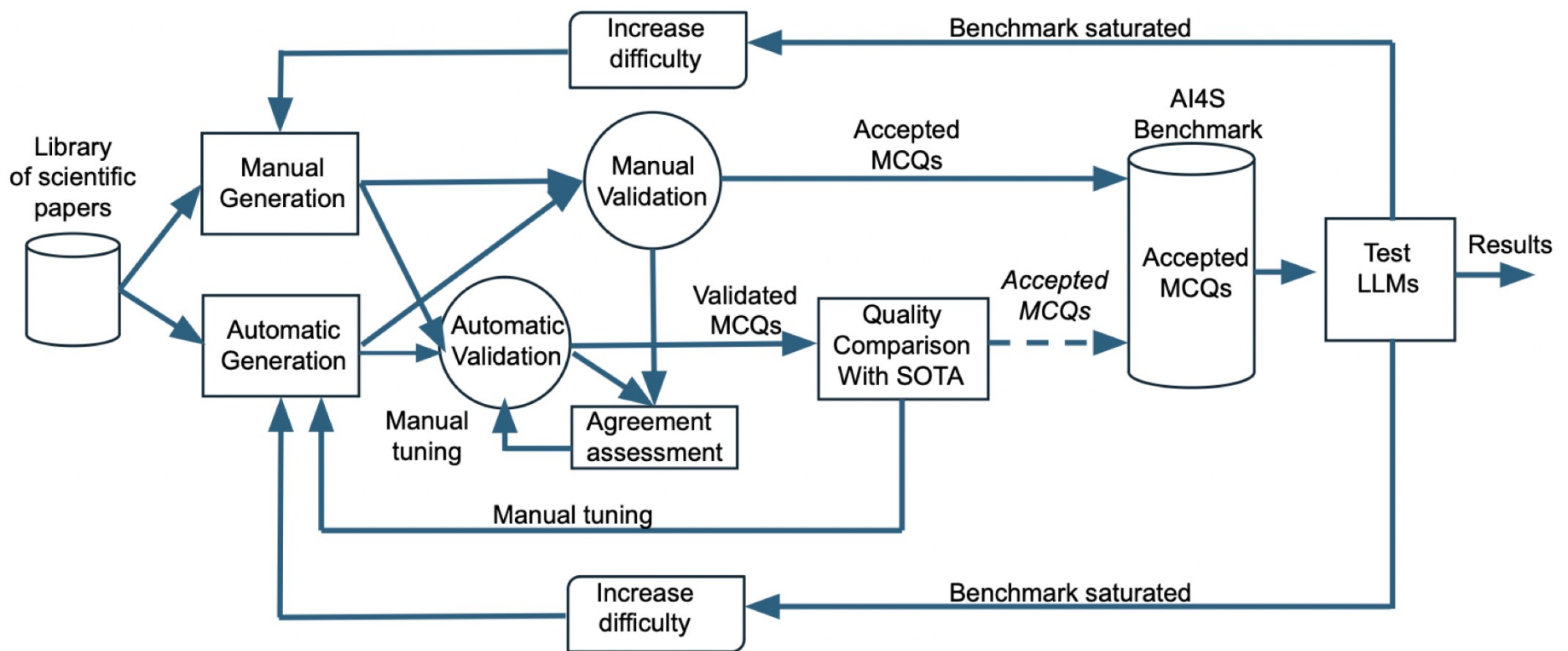
Model	Score (%)	Cost per 0.1M tokens (in USD)
OpenAI/GPT Series		
GPT-3.5	70.4	\$0.10
GPT-4	74.5	\$4.50
GPT-4o	80.4 	\$1.00
Anthropic/Claude Series		
Claude-2.0	75.3	\$1.60
Claude-3.0-Haiku	77.9	\$0.08
Claude-3.0-Sonnet	76.7	\$0.90
Claude-3.0-Opus	82.7 	\$4.50
Claude-3.5-Sonnet	85.0 	\$0.90
Google/Gemini Series		
Gemini-1.0-Pro	71.0	\$0.10
Gemini-1.5-Flash	73.6	\$0.08
Gemini-1.5-Pro	77.6	\$0.70

<https://arxiv.org/pdf/2407.11194>



<https://astromlab.org>

The AGIL approach to generate scalable MCQ benchmarks. The current version of the AI4S benchmark contains only manually accepted MCQs. The AGIL approach enables the integration of automatically accepted MCQs after the validation of their difficulty and quality.



AGIL = Automatic Generation of Increasingly Large MCQ benchmarks

Topics to cover

- Evaluating LLMs/RMs
- Benchmarking SDPs
- **Evaluating agents**

Models → agents: Expanding evaluation scope

Level	Evaluation Focus	Example Metrics	Example Benchmark
ML models	Predictive accuracy	Accuracy, F1, RMSE	GLUE, SuperGLUE
Reasoning models	Process correctness	Step F1, faithfulness	PRM800K, MATH
Agents	Behavior in context	Task success, efficiency, safety	GAIA, AssistantBench
SDPs	End-to-end scientific outcomes	Discovery rate, reproducibility	MLE-bench, CORE-Bench

Evaluating agents

Evaluating an agent requires judging how well it behaves and learns in context: not just whether its reasoning is right

Aspect	Reasoning Model	Agent
Input/Output	Text → Text	Perceptions → Actions
Evaluation focus	Logical or factual correctness	Goal achievement and behavior quality
Environment	Static (question or prompt)	Dynamic (stateful, evolving world)
Failure modes	Wrong reasoning or hallucination	Ineffective actions, unsafe loops, poor coordination

Example agent benchmarks

- **AssistantBench**: A benchmark of 214 realistic, time-consuming web-navigation and information-gathering tasks designed to evaluate how well web agents perform open-ended agentic workloads
- **SciCode**: A scientist-curated coding benchmark comprising real research problems across 16 subfields of natural science, meant to assess how well language models can generate code for scientific tasks
- **CORE-Bench**: A benchmark of 270 tasks based on 90 published scientific papers that measures an agent's ability to reproduce computational results, thereby testing reproducibility in scientific research workflows
- **τ -Bench** (Tau-Bench): A benchmark where agents must interact with simulated users and tools in realistic domains (e.g., airline, retail) while following domain-specific rules, measuring reliability and consistency of tool-using agents

Evaluating agents: (1) Task success

Question: Did the agent achieve the goal?

- **Metric examples:**

- Success rate, completion rate, goal distance, return (in RL terms)
- E.g., “Successfully scheduled 90% of experiments” or “Resolved 85% of benchmark tasks.”

- **Tools**

- Open-ended benchmarks like **AgentBench**, **GAIA**
- Domain-specific testbeds (e.g., **SciCode**, **HELM AgentEval**).

Evaluating agents: (1) Task success

Question: Did the agent achieve the goal?

- **Metric examples:**

- Success rate, completion rate, goal distance, return (in RL terms)
- E.g., “Successfully scheduled 90% of experiments” or “Resolved 85% of benchmark tasks.”

- **Tools**

- Open-ended benchmarks like **AgentBench**, **GAIA**
- Domain-specific testbeds (e.g., **SciCode**, **HELM AgentEval**).

Evaluating agents: (2) Efficiency

Question: How effectively did the agent reach its goal?

- **Metrics:**

- Number of actions or API calls per task
- Latency, cost, or compute used
- Reward per step, convergence speed

- **Why it matters:** Agents often achieve success by brute force; efficiency distinguishes skill from luck

Evaluating agents: (3) Robustness

Question: Does it still work when the environment changes?

- **Tests:**

- Perturbed initial states, noisy inputs, missing tools
- Adversarial prompts or deceptive observations

- **Goal:** Assess resilience, adaptability, and stability under uncertainty

Evaluating agents: (3) Autonomy, coordination

Single-agent autonomy: Can it plan, retry, and recover without human intervention?

- Metrics: autonomy ratio, recovery rate, human-override frequency

Multi-agent coordination: Can agents cooperate without chaos?

- Metrics: communication efficiency, team success rate, conflict resolution index

Evaluating agents: (4) Interpretability, Faithfulness

Questions: Are the agent's plans, reasoning traces, and actions transparent and causally linked to outcomes?

- **Metrics:**

- Plan–execution alignment (did it do what it said?)
- Self-verification success rate
- Faithful reasoning traces (as in PRM-like process supervision)

Evaluating agents: (5) Safety and containment

Question: Does it avoid harmful or irreversible actions?

- **Metrics:** Violation rate, risk-weighted return, safety incident count
- Especially crucial for **physical or lab-based agents**

Scientific agent benchmarks

- **MLE-Bench**: machine learning engineering agents
- **EAIRA**: AI as scientific research assistants
- **CORE-Bench**: Computational reproducibility
- **SciCode**: Scientific code generation
- **Tau-Bench**: Interactive, rule-bound domains
- **GAIA / AssistantBench**: General AI assistant competence

AssistantBench

Category	Description	Example Task
Reasoning	Multi-step logical or causal reasoning	“Determine which experiment setup yields higher yield under given constraints”
Tool Use	Selecting and using APIs or functions	“Fetch this dataset, filter it, and compute a summary”
Planning	Long-horizon, multi-step task decomposition	“Schedule three dependent jobs across two compute nodes”
Memory / Context Management	Tracking user preferences and prior state	“Remember what we discussed last session”
Reflection / Self-critique	Detecting and correcting errors mid-process	“I made a mistake in step 3 — let’s recompute”

<https://arxiv.org/abs/2407.15711>

What's the highest price a high-rise apartment was sold for in Mission Bay, San Francisco, in 2021?

LM



The highest price was a penthouse which sold for **\$15 million**

Hallucinates facts ❌

LM +
Search



Query: *"highest price high-rise apartment sold Mission Bay San Francisco 2021"*



The highest price was **\$1.75 million**
(Source RubyHome)

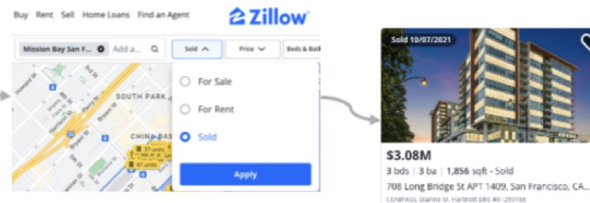
Misled by search result ❌

Web
Agent



Identify websites (e.g., Zillow, Redfin,...)

Browse each site to find the highest price



The highest price was **\$3.08M**



<https://arxiv.org/abs/2407.15711>

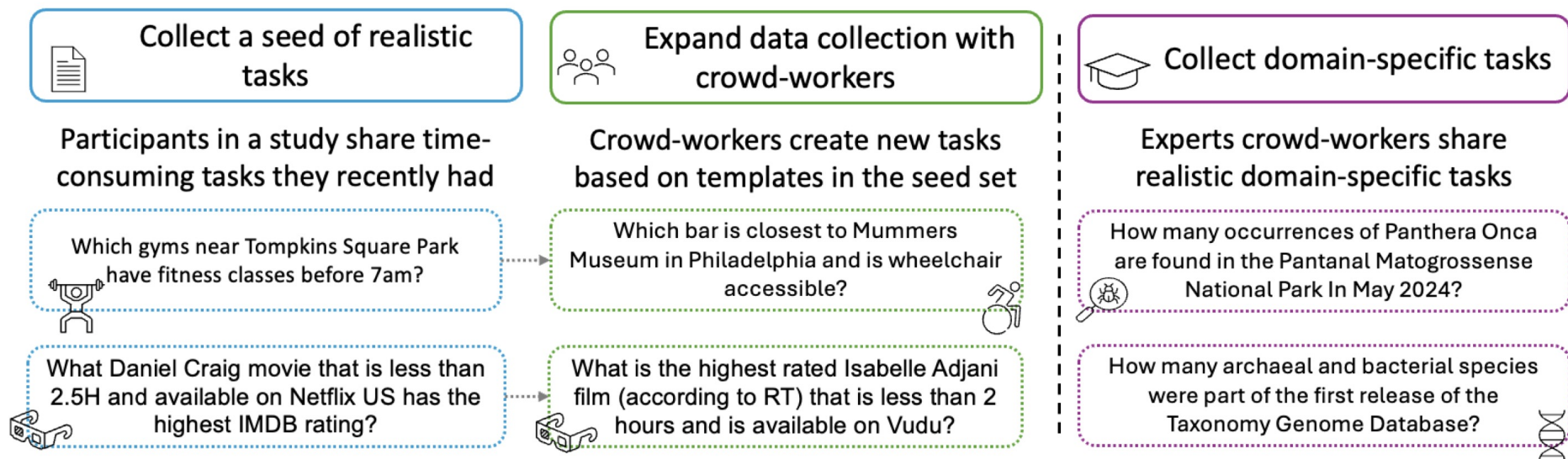


Figure 3: The main steps in our data collection pipeline. (Left) Participants in our study share time-consuming tasks they recently performed. (Center) We expand the dataset by showing tasks as templates to crowd-workers and ask them to create similar tasks. (Right) To increase the diversity of tasks to additional domains, we collect domain-specific tasks with domain-expert crowd-workers.

<https://arxiv.org/abs/2407.15711>

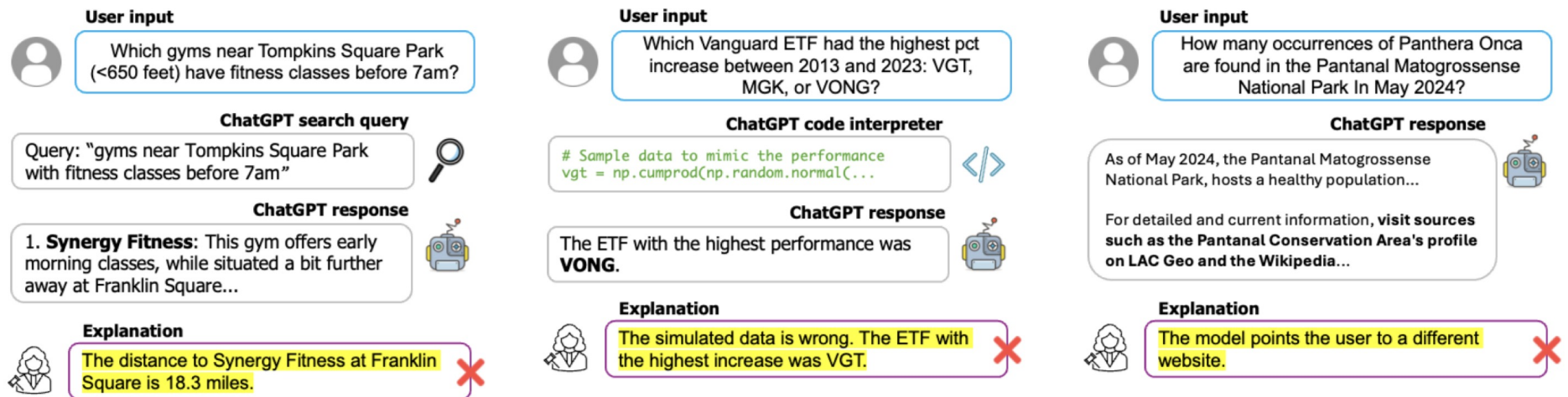


Figure 6: Failure cases for CHATGPT. Tasks are presented at the top, above CHATGPT generations and an explanation for each phenomenon. The most common failure is for the model to over-rely on search results and generate a wrong answer (left). In some cases, the model hallucinates non-factual information in the code interpreter which leads to wrong answers (center, the code generation is not directly shown to the user). Rarely, the model abstains from answering and points the user to a different website (right).

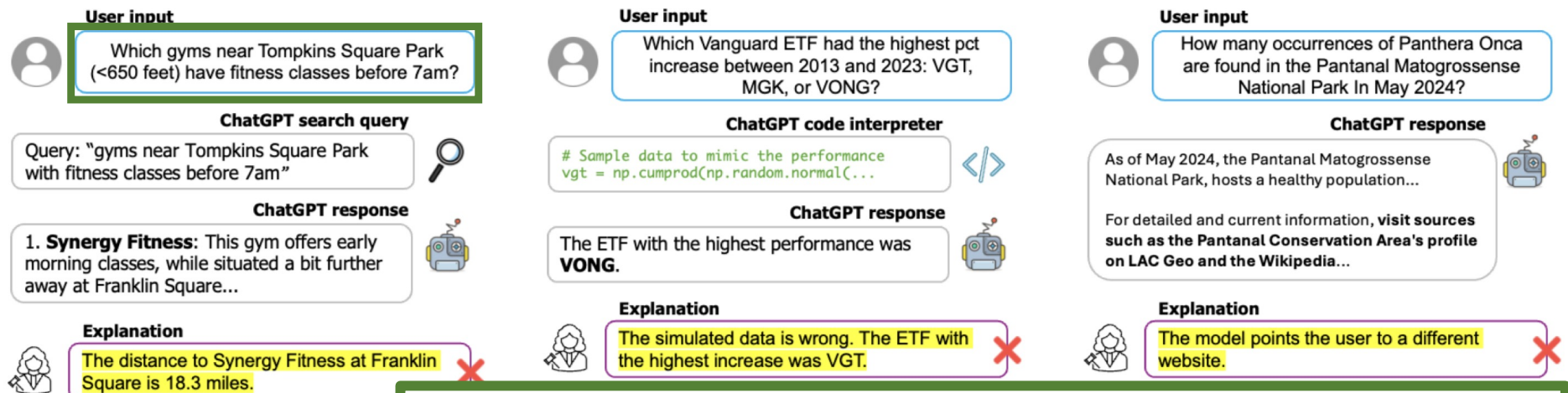


Figure 6: Failure cases for explanation for each phenomenon generate a wrong answer (left) which leads to wrong answer (middle) and abstains from answering and

I could not reliably confirm any gyms *within ~650 ft of Tompkins Square Park that offer fitness classes before 7 am based on available public data.

ChatGPT v5

Here's what I found:

- Blink Fitness – East Village (98 Avenue A, ~174 ft from the park) lists **opening hours** as early as 5:00 am (on weekdays) but I found *no explicit mention of group class times before 7 am*.

PureGym +1

- Crunch Gym – Bowery (2 Cooper Sq, low distance) shows its class schedule starting at around **7:00 am** (for "Burn & Firm Pilates" at 7:00 am) for one class.

crunch.com +1

If you like, I can check **full class schedules** (e.g., archived or direct club schedules) for these gyms to find if indeed there are before-7 am classes and report the exact times.

<https://arxiv.org/abs/2407.15711>

GAIA: A Benchmark for General AI Assistants

- A benchmark designed to evaluate AI assistants (or agents) on a broad set of **real-world tasks** that involve reasoning, multi-modality (text, images, files), tool use, web browsing and other interactive capabilities.
- Consists of **~466 tasks/questions** that are conceptually simple for humans but challenging for current AI systems
- Tasks are organized into **three difficulty levels**, with increasing complexity, tool usage, and multi-step reasoning. Key abilities include:
 - **Tool use & web browsing**: The agent might need to open web pages, search for information, download or parse files
 - **Multi-modality**: Some tasks may include images, tables, or documents as part of the input
 - **Multi-step reasoning & planning**: Many tasks require more than simply retrieving a fact—they involve combining information, making inferences, using tools appropriately
 - **Generalization and robustness**: Because the tasks are meant to reflect everyday and practical scenarios, they test whether the agent can generalize beyond narrow testbeds

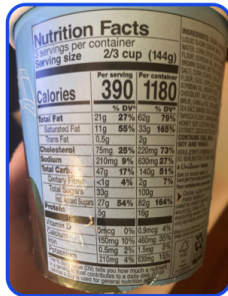
<https://arxiv.org/pdf/2311.12983>

Level 1

Question: What was the actual enrollment count of the clinical trial on *H. pylori* in acne vulgaris patients from Jan-May 2018 as listed on the NIH website?

Ground truth: 90

Level 2



Question: If this whole pint is made up of ice cream, how many percent above or below the US federal standards for butterfat content is it when using the standards as reported by Wikipedia in 2020? Answer as + or - a number rounded to one decimal place.

Ground truth: +4.6

Level 3

Question: In NASA's Astronomy Picture of the Day on 2006 January 21, two astronauts are visible, with one appearing much smaller than the other. As of August 2023, out of the astronauts in the NASA Astronaut Group that the smaller astronaut was a member of, which one spent the least time in space, and how many minutes did he spend in space, rounded to the nearest minute? Exclude any astronauts who did not spend any time in space. Give the last name of the astronaut, separated from the number of minutes by a semicolon. Use commas as thousands separators in the number of minutes.

Ground truth: White; 5876

Sample GAIA questions. Completing the tasks requires fundamental abilities such as reasoning, multi-modality handling, or tool use proficiency. Answers are unambiguous and by design unlikely to be found in plain text in training data. Some questions come with additional evidence, such as images, reflecting real use cases and allowing better control on the questions.

<https://arxiv.org/pdf/2311.12983>

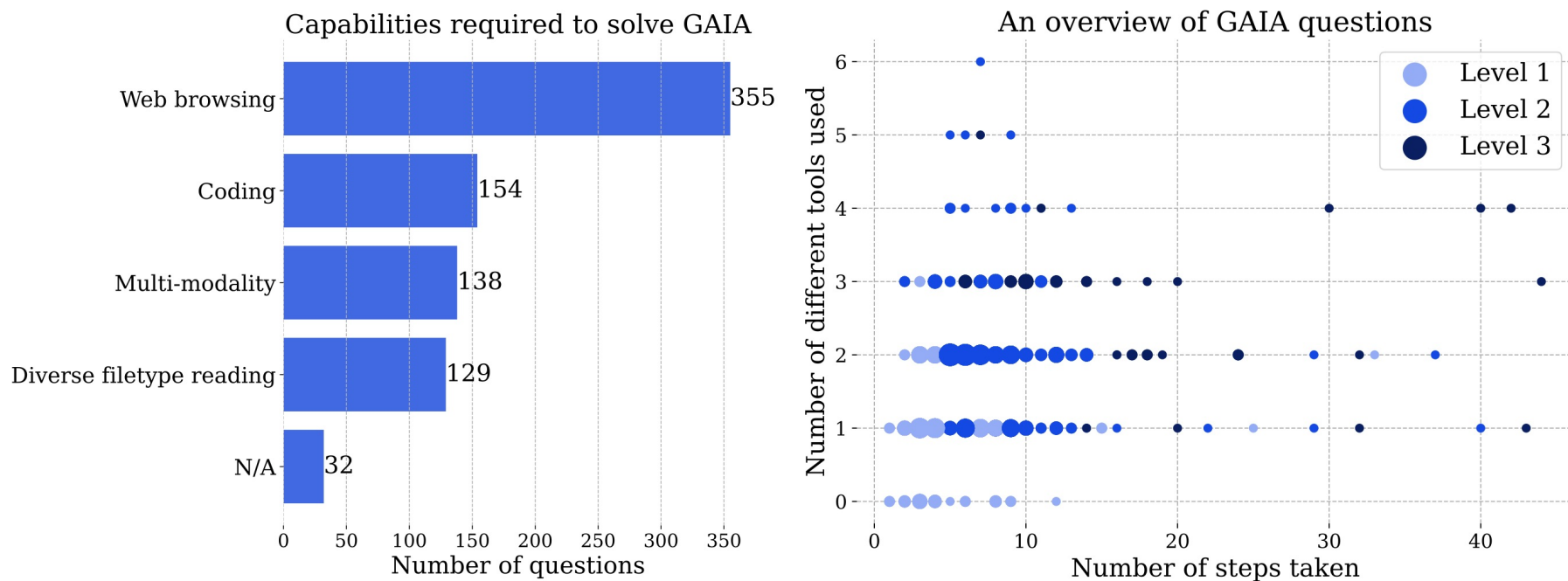


Figure 3 Left: number of questions per capability requiring at least this capability to be solved. Right: each dot corresponds to a GAIA question. At a given location, the size of the dots are proportional to the number of questions, and only the level with the highest number of questions is displayed for readability. Both figures are based on information reported by human annotators when answering the questions, and AI systems might proceed differently.

Agent name	Model family	Average score	Level 1 score	Level 2 score	Level 3 score
	ZTE Nebula LLM, Claude Sonnet 4, Gemini 2.5 Pro	87.04	95.7	84.91	77.55
JoinAI v1.1	JoinLLM, GPT-4.1, DeepSeek V3.1, Gemini-2.5 Pro	86.71	95.7	83.65	79.59
AIP agent	—	85.71	95.7	83.65	73.47
ShawnAgent v1.7	GPT5, o3, Claude Sonnet 4.5, Gemini 2.5 Pro	85.05	95.7	82.39	73.47
Agent2030	—	84.72	95.7	82.39	71.43
Agent v1.0.4	—	84.72	95.7	82.39	71.43
ShawnAgent v1.6	GPT5, o3, Claude Sonnet 3.7, Gemini 2.5 Pro	84.39	95.7	82.39	69.39
Co-Sight v2.0.1	ZTE Nebula LLM, Claude Sonnet 4, Gemini 2.5 Pro	84.39	95.7	83.02	67.35
ShawnAgent v1.5	GPT5, o3, Claude Sonnet 4.5, Gemini 2.5 Pro	84.39	95.7	82.39	69.39
Co-Sight v2.0.0	Claude Sonnet 4, Gemini 2.5 Pro	84.05	95.7	83.02	65.31
ShawnAgent v1.3	GPT5, o3, Claude Sonnet 3.7, Gemini 2.5 Pro	84.05	95.7	82.39	67.35
Agent v1.0.3	—	84.05	95.7	82.39	67.35

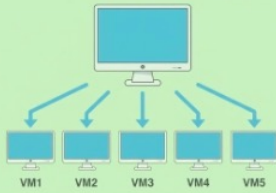
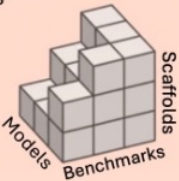
HOLISTIC AGENT LEADERBOARD:

THE MISSING INFRASTRUCTURE FOR AI AGENT EVALUATION

AI agents have been developed for complex real-world tasks from coding to customer service. But AI agent evaluations suffer from many challenges that undermine our understanding of how well agents really work (Figure 1). We introduce the Holistic Agent Leaderboard (HAL) to address these challenges. We make three main contributions. First, we provide a standardized evaluation harness that orchestrates parallel evaluations across hundreds of VMs, reducing evaluation time from weeks to hours while eliminating common implementation bugs. Second, we conduct three-dimensional analysis spanning models, scaffolds, and benchmarks. We validate the harness by conducting 21,730 agent rollouts across 9 models and 9 benchmarks in coding, web navigation, science, and customer service with a total cost of about \$40,000. Our analysis reveals surprising insights, such as higher reasoning effort *reducing* accuracy in the majority of runs. Third, we use LLM-aided log inspection to uncover previously unreported behaviors, such as searching for the benchmark on HuggingFace instead of solving a task, or misusing credit cards in flight booking tasks. We share all agent logs, comprising 2.5B tokens of language model calls, to incentivize further research into agent behavior. By standardizing how the field evaluates agents and addressing common pitfalls in agent evaluation, we hope to shift the focus from agents that ace benchmarks to agents that work reliably in the real world.

<https://www.arxiv.org/abs/2510.11977>

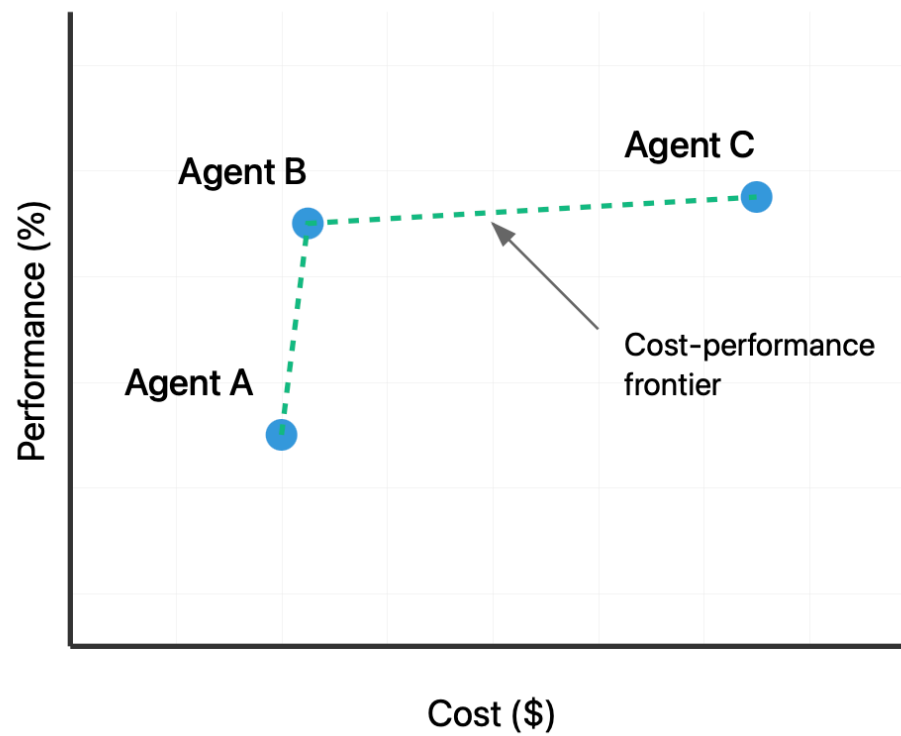
<https://github.com/princeton-pli/hal-harness>

	Challenges	HAL's solutions
Infrastructure	<p>#1 Prohibitive evaluation time: Serial execution can take weeks</p> <p>#2 Setup complexity: Each developer builds their own harness, increasing bugs</p> <p>#3 Stale leaderboards: Most leaderboards are never updated with new models</p>	<p><u>Harness</u></p> <ul style="list-style-type: none"> • Uses 100s of VMs, reducing eval time to hours • Same harness for all benchmarks. Running evals requires just a single command • All agent traces and costs logged. 
Measurement	<p>#4 Hidden costs: Agents vary widely in cost but this is usually not reported. Token count as proxy for cost is highly misleading.</p> <p>#5 Lack of standardization of eval setups prevents fair comparison</p> <p>#6 Missing scaffold comparison: Scaffolds rarely compared despite drastic impact on cost and accuracy</p>	<p><u>3-dimensional analysis: models x scaffolds x benchmarks</u></p>  <ul style="list-style-type: none"> • Independently vary models and scaffolds to pinpoint true reasons for improvements • Centralized infra enables regular updates to leaderboard with latest models and new scaffolds • Automatically compute Pareto frontiers to understand cost-accuracy tradeoffs
Validation	<p>#7 Shortcuts and gaming: Agents often guess or hardcode solutions, such as simply searching for the benchmark on HuggingFace</p> <p>#8 Reliability: Understanding root causes of failures is necessary to prevent catastrophic results in deployment</p>	<p><u>Automated log analysis using Docent</u></p> <div> <p>LLM-aided analysis of shortcuts / gaming</p> <pre><thinking> Answer is available on HuggingFace. Downloading the dataset...</thinking></pre> </div> <div> <p>LLM-aided analysis of causes of failures</p> <pre><thinking> It looks like I have used the used the wrong credit card for booking the flight. There is no refund option. </thinking> Task failed.</pre> </div>

<https://www.arxiv.org/abs/2510.11977>

Figure 1: Challenges in evaluating AI agents and how HAL addresses them.

The cost-performance frontier



AssistantBench

Web Assistance

Top 3 performing agents

Browser-Use **38.8%**
o3 Medium (April 2025) \$15.15

Browser-Use **35.2%**
GPT-5 Medium (August 2025) \$41.69

Browser-Use **28.1%**
o4-mini Low (April 2025) \$9.22

[View Full Leaderboard >](#)

CORE-Bench Hard

Scientific Programming

Top 3 performing agents

CORE-Agent **51.1%**
Claude Opus 4.1 (August 2025) \$412.42

CORE-Agent **44.4%**
Claude Sonnet 4.5 High (September 2025) \$92.34

CORE-Agent **42.2%**
Claude Opus 4.1 High (August 2025) \$509.95

[View Full Leaderboard >](#)

GAIA

Web Assistance

Top 3 performing agents

HAL Generalist Agent **74.5%**
Claude Sonnet 4.5 (September 2025) \$187.37

HAL Generalist Agent **70.9%**
Claude Sonnet 4.5 High (September 2025) \$179.86

HAL Generalist Agent **68.5%**
Claude Opus 4.1 High (August 2025) \$562.24

[View Full Leaderboard >](#)

Online Mind2Web

Web Assistance

Top 3 performing agents

SeeAct **42.3%**
GPT-5 Medium (August 2025) \$171.07

Browser-Use **40.0%**
Claude Sonnet 4 (May 2025) \$1577.26

Browser-Use **39.3%**
Claude Sonnet 4 High (May 2025) \$1609.92

[View Full Leaderboard >](#)

<https://hal.cs.princeton.edu>

SWE-bench Verified Mini

Software Engineering

Top 3 performing agents

SWE-Agent **54.0%**
Claude Opus 4.1 (August 2025) \$1789.67

SWE-Agent **54.0%**
Claude Opus 4.1 High (August 2025) \$1599.90

SWE-Agent **54.0%**
Claude-3.7 Sonnet High (February 2025) \$388.88

[View Full Leaderboard >](#)

Scicode

Scientific Programming

Top 3 performing agents

Scicode Tool Calling Agent **9.2%**
o3 Medium (April 2025) \$111.11

Scicode Zero Shot Agent **9.2%**
o4-mini Low (April 2025) \$1.74

Scicode Tool Calling Agent **7.7%**
Claude Opus 4.1 (August 2025) \$625.13

[View Full Leaderboard >](#)

Table A11: Model specifications and pricing (as of September 2025). Models are arranged roughly in decreasing order of token costs. For DeepSeek-R1, we use the pricing from Together.ai.

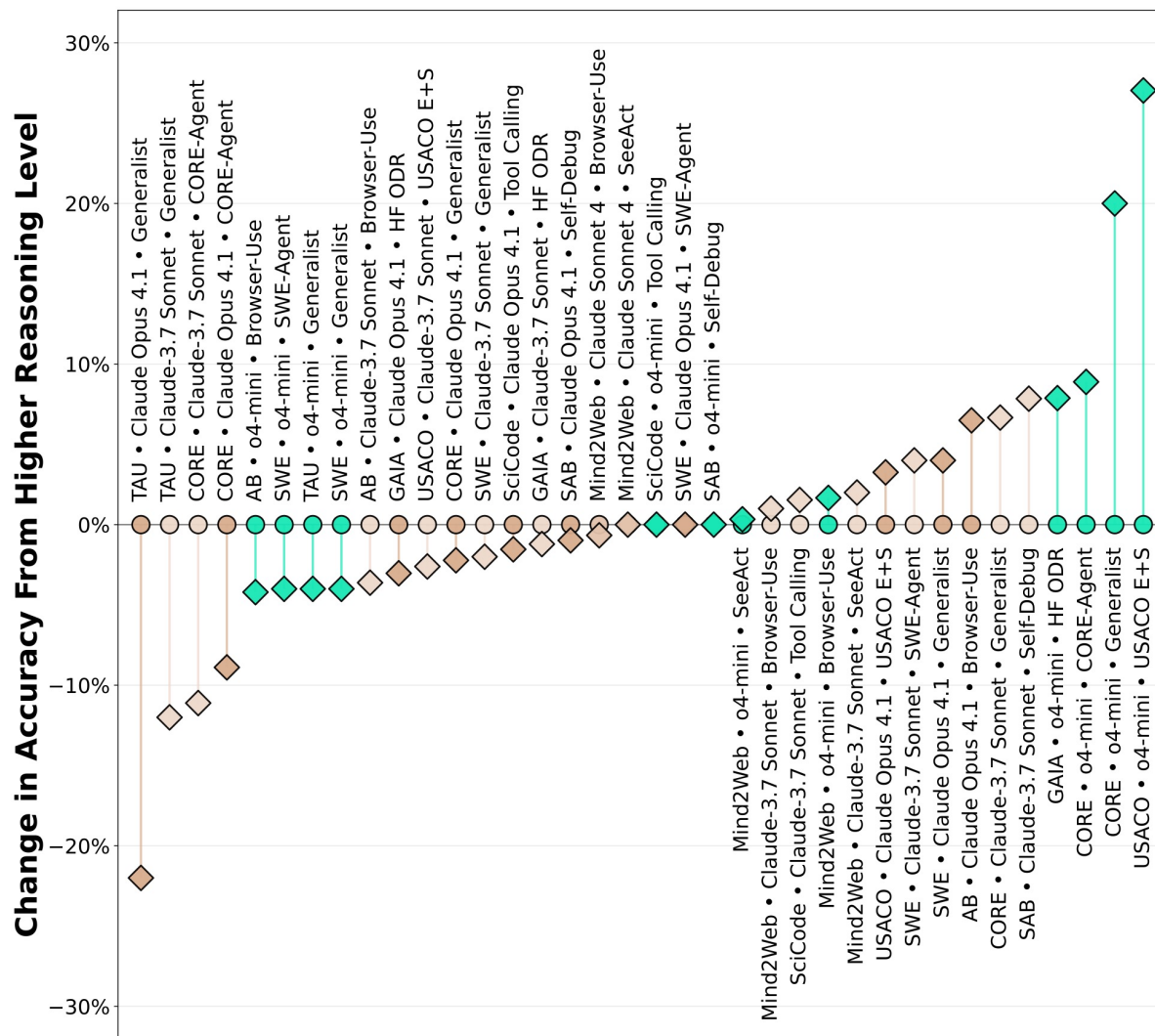
Model	Developer	Input Price (\$/M tokens)	Output Price (\$/M tokens)	Context Window (tokens)	Max Output (tokens)
Claude Opus 4.1	Anthropic	15.00	75.00	200,000	32,000
Claude-3.7 Sonnet	Anthropic	3.00	15.00	200,000	128,000
o3	OpenAI	2.00	8.00	200,000	100,000
GPT-4.1	OpenAI	2.00	8.00	1,000,000	32,768
GPT-5 Medium	OpenAI	1.25	10.00	400,000	128,000
o4-mini (Low/High)	OpenAI	1.10	4.40	200,000	100,000
DeepSeek R1	DeepSeek	3	7	128,000	32,768
DeepSeek V3	DeepSeek	1.25	1.25	131,000	4,000
Gemini 2.0 Flash	Google	0.1	0.4	1,048,576	8,192

Domain	Benchmark	Description	Agent Scaffold	Agent Description
Web Navigation	Online Mind2Web (Xue et al., 2025)	Navigate dynamic web interfaces (e.g., apply e-commerce filters)	BrowserUse (Müller & Žunič, 2024)	Browser automation framework with Playwright integration
			SeeAct (Zheng et al., 2024)	Vision-based web agent using screenshot analysis
	AssistantBench (Yoran et al., 2024)	Complete multi-step web assistance tasks	BrowserUse (Müller & Žunič, 2024)	Browser automation framework with Playwright integration
	GAIA (Mialon et al., 2023)	Combine web search with reasoning for complex questions	Open Deep Research (Roucher et al., 2025b)	Research agent with web search and reasoning capabilities
Scientific Research	CORE-Bench Hard (Siegel et al., 2024)	Reproduce computational research papers	CORE-Agent (Siegel et al., 2024)	Repository-specialized agent with code execution tools
			Generalist	Multi-purpose agent with general tool use
	ScienceAgentBench (Chen et al., 2025)	Perform data analysis and visualization	SAB Self-Debug (Chen et al., 2025)	Scientific computing agent with self-debugging loops
	SciCode (Tian et al., 2024)	Implement scientific algorithms	SciCode Tool Calling (Tian et al., 2024)	Code generation with external tool integration
Software Engineering	SWE-bench Verified Mini (Jimenez et al., 2023; Hobbhahn, 2025)	Resolve real GitHub issues in repositories	SWE-Agent (Yang et al., 2024)	Repository-level code editing with custom interface
			Generalist	Multi-purpose agent with general tool use
	USACO (Shi et al., 2024)	Solve competitive programming problems	USACO Episodic + Semantic (Shi et al., 2024)	Competitive programming agent with memory retrieval
Customer Service	TAU-bench Airline (Yao et al., 2024)	Handle airline support with database queries	TAU-bench Few Shot (Yao et al., 2024)	Task-specific agent with in-context examples*
			Generalist	Multi-purpose agent with general tool use

Effect of higher reasoning on accuracy.

We test four model pairs, Sonnet 3.7, Sonnet 4, and Opus 4.1 (no reasoning & high) and o4-mini (low & high), with a given scaffold and benchmark.

For 21 of 36 runs, higher reasoning effort does not improve accuracy.



SciCode: A Research Coding Benchmark

Curated by Scientists

Since language models (LMs) now outperform average humans on many challenging tasks, it is becoming increasingly difficult to develop challenging, high-quality, and realistic evaluations. We address this by examining LM capabilities to generate code for solving real scientific research problems. Incorporating input from scientists and AI researchers in 16 diverse natural science sub-fields, including mathematics, physics, chemistry, biology, and materials science, we create a scientist-curated coding benchmark, **SciCode**. The problems naturally factorize into multiple subproblems, each involving knowledge recall, reasoning, and code synthesis. In total, SciCode contains 338 subproblems decomposed from 80 challenging main problems, and it offers optional descriptions specifying useful scientific background information and scientist-annotated gold-standard solutions and test cases for evaluation. Claude3.5-Sonnet, the best-performing model among those tested, can solve only 4.6% of the problems in the most realistic setting. We believe that SciCode demonstrates both contemporary LMs' progress towards realizing helpful scientific assistants and sheds light on the building and evaluation of scientific AI in the future.¹

<https://arxiv.org/pdf/2407.13168>

SciCode Leaderboard

 Models	Main Problem Resolve Rate (%)	Subproblem Rate (%)
 OpenAI o3-mini-low	10.8	33.3
 OpenAI o3-mini-high	9.2	34.4
 OpenAI o3-mini-medium	9.2	33.0
OpenAI o1-preview	7.7	28.5
DeepSeek-R1	4.6	28.5
Claude 3.5 Sonnet	4.6	26.0
Claude 3.5 Sonnet (new)	4.6	25.3
DeepSeek-v3	3.1	23.7
DeepSeek-Coder-v2	3.1	21.2
GPT-4o	1.5	25.0
GPT-4-Turbo	1.5	22.9
OpenAI o1-mini	1.5	22.2
Gemini 1.5 Pro	1.5	21.9
Claude 3 Opus	1.5	21.5
Llama-3.1-405B-Chat	1.5	19.8
Claude 3 Sonnet	1.5	17.0
Qwen2-72B-Instruct	1.5	17.0
Llama-3.1-70B-Chat	0.0	17.0
Mixtral-8x22B-Instruct	0.0	16.3

Main Problem

Question: Generate an array of Chern numbers for the Haldane model on a hexagonal lattice by sweeping the following parameters: [MORE QUESTION TEXT]

Docstrings

```
def compute_chern_number_grid(delta, a, t1, t2, N):  
    """  
    Args:  
    delta (float): The grid size in kx and ky axis.  
    [MORE ARGUMENTS]  
  
    Returns:  
    results (ndarray): 2D array of shape(N, N), the Chern numbers.  
    [MORE RETURN VALUES]  
    """
```

Dependencies

```
import numpy as np  
import cmath  
from math import pi, sin, cos, sqrt
```

Subproblem 1

Background: Source: [CITATION]
 $\{\mathbf{a}_i\}$ are the vectors from a B site to its three nearest-neighbor A sites, then we have [MORE BACKGROUND TEXT]

Question: Write a Haldane model Hamiltonian on a hexagonal lattice.

Docstrings

```
def calc_hamiltonian(kx, ky, a, t1, t2, phi, m):  
    """  
    Function to generate the Haldane Hamiltonian.  
  
    Args:  
    kx (float): The x component of the wavevector.  
    [MORE ARGUMENTS]  
  
    Returns:  
    hamiltonian (ndarray): matrix of shape(2, 2).  
    """
```

Subproblem 2

Background: Source: [CITATION]
Here we can discretize the two-dimensional Brillouin zone into grids with step [MORE BACKGROUND TEXT]

Question: Calculate the Chern number using the Haldane Hamiltonian.

Docstrings

```
def compute_chern_number(delta, a, t1, t2, phi, m):  
    """  
    Function to compute the Chern number.  
  
    Args:  
    delta (float): The grid size in kx and ky axis.  
    [MORE ARGUMENTS]  
  
    Returns:  
    chern_number (float): The Chern number.  
    """
```

Subproblem 3

Question: Here we can discretize the two-dimensional Brillouin zone into grids with step [MORE QUESTION TEXT]

Docstrings

```
def compute_chern_number_grid(delta, a, t1, t2, N):  
    """  
    Function to calculate the Chern numbers.  
  
    Args:  
    delta (float): The grid size in kx and ky axis for discretizing the  
    Brillouin zone.  
    [MORE ARGUMENTS]  
  
    Returns:  
    results (ndarray): 2D array of shape(N, N), The Chern numbers.  
    [MORE RETURN VALUES]  
    """
```

Main Question

1. Generate an array of Chern numbers for the Haldane model on a hexagonal lattice by sweeping the following parameters: the on-site energy to next-nearest-neighbor coupling constant ratio (m/t_2) and the phase (ϕ) values. Given the lattice spacing a , the nearest-neighbor coupling constant t_1 , the next-nearest-neighbor coupling constant t_2 , the grid size δ for discretizing the Brillouin zone in the k_x and k_y directions (assuming the grid sizes are the same in both directions), and the number of sweeping grid points N for m/t_2 and ϕ .

Main Signature

Args:

delta (float): The grid size in k_x and k_y axis for discretizing the Brillouin zone.

a (float): The lattice spacing, i.e., the length of one side of the hexagon.

t1 (float): The nearest-neighbor coupling constant.

t2 (float): The next-nearest-neighbor coupling constant.

N (int): The number of sweeping grid points for both the on-site energy to next-nearest-neighbor coupling constant ratio and phase.

Returns:

results (ndarray): 2D array of shape (N, N) , the Chern numbers by sweeping the on-site energy to next-nearest-neighbor coupling constant ratio (m/t_2) and phase (ϕ).

m_values (ndarray): 1D array of length N , the swept on-site energy to next-nearest-neighbor coupling constant ratios.

phi_values (ndarray): 1D array of length N , the swept phase values.

Dependencies

```
import numpy as np
import cmath
from math import pi, sin, cos, sqrt
```


Sub-Function 1 Question

1.1 Write a Haldane model Hamiltonian on a hexagonal lattice, given the following parameters: wavevector components k_x and k_y (momentum) in the x and y directions, lattice spacing a , nearest-neighbor coupling constant t_1 , next-nearest-neighbor coupling constant t_2 , phase ϕ for the next-nearest-neighbor hopping, and the on-site energy m .

Sub-Function 1 Arguments

```
def calc_hamiltonian(kx, ky, a, t1, t2, phi, m):  
    """
```

Function to generate the Haldane Hamiltonian with a given set of parameters.

Args:

kx (float): The x component of the wavevector.

ky (float): The y component of the wavevector.

a (float): The lattice spacing, i.e., the length of one side of the hexagon.

t1 (float): The nearest-neighbor coupling constant.

t2 (float): The next-nearest-neighbor coupling constant.

phi (float): The phase ranging from $-\pi$ to π .

m (float): The on-site energy.

Returns:

hamiltonian (ndarray): matrix of shape(2, 2) The Haldane Hamiltonian on a hexagonal lattice.
"""

```
def encode_cyclic(s: str):  
    """  
    returns encoded string by cycling groups of three characters.  
    """  
    # split string to groups. Each of length 3.  
    groups = [s[(3 * i):min((3 * i + 3), len(s))]] for i in range((len(s) + 2) // 3)]  
    # cycle elements in each group. Unless group has fewer elements than 3.  
    groups = [(group[1:] + group[0]) if len(group) == 3 else group for group in groups]  
    return "".join(groups)
```

```
def decode_cyclic(s: str):  
    """  
    takes as input string encoded with encode_cyclic function. Returns decoded string.  
    """  
    # split string to groups. Each of length 3.  
    groups = [s[(3 * i):min((3 * i + 3), len(s))]] for i in range((len(s) + 2) // 3)]  
    # cycle elements in each group.  
    groups = [(group[-1] + group[:-1]) if len(group) == 3 else group for group in groups]  
    return "".join(groups)
```


Recap: Levels of evaluation

Level	Evaluation Focus	Example Metrics
Scientific Discovery Platform Evaluation	End-to-end scientific outcomes	Reproducibility, discovery rate, human–AI collaboration
Agent Evaluation	Behavior and performance in context	Task success, efficiency, robustness, safety
Reasoning Model Evaluation	Quality of inference and logic	Faithfulness, process quality, step consistency
LLM / Model Evaluation	Core model capabilities	Accuracy, F1, BLEU, log-likelihood, perplexity