

Closed-loop experimentation and hypothesis revision

Data validation, belief revision and search-value updates.

Reasoning agents, feedback, and scientific evidence

A failed run has more than one explanation

A research run returns no useful result. Perhaps the hypothesis failed, the instrument failed, or the program never reached the measurement.

These outcomes call for different changes. Treating every failure as evidence against the hypothesis can steer the investigation away from a sound explanation.

The opportunity is to make feedback identify what should change, rather than merely trigger another attempt.

Locating what feedback should change

Locate the failed commitment. An import error concerns execution, a failed reference concerns measurement, and a valid conflicting observation concerns the scientific model.

Now construct the smallest defensible revision: repair the procedure, revise a parameter, change the mechanism or obtain another measurement.

A closed loop results when the next decision depends on this new information. Repetition alone does not specify which object learned from the feedback.

Choosing a revision after feedback

Candidate	Reason to choose it	Cost or limitation
Repeat or repair execution	No admissible scientific observation was produced	Repeating a valid result adds little independent evidence
Update a parameter or belief	The model can represent the new observation	A flexible fit can hide a missing mechanism
Expand the model or proposal	A tested assumption cannot explain the result	Extra flexibility needs discriminating evidence

Choose the revision from the origin and reliability of the feedback. Preserve uncertainty when several failure explanations remain possible.

Pedagogical reconstruction

[Chaloner and Verdinelli \(1995\)](#)

[Agrawal et al. \(2025\)](#)

Validate the observation before updating the mechanism

A failed procedure, missing input, or corrupted measurement is evidence about execution reliability. A valid measurement that contradicts a prediction is evidence about the scientific model.

Both matter, but they justify different updates. Record the failure type and the observation's relationship to the intended experiment.

Otherwise the system can weaken a good hypothesis because a tool failed or repeatedly repair code when the scientific prediction was simply wrong.

Bayesian updating preserves graded uncertainty

$$\text{Posterior odds} = \text{prior odds} \times \frac{p(y | A, D)}{p(y | B, D)}$$

Posterior odds equal prior odds multiplied by the likelihood ratio for the new observation. The likelihood must condition on relevant previous evidence.

Reusing the same source as if it were a new independent observation can create unjustified confidence.

In an open-ended hypothesis space, updating probabilities within the current model class may need to be complemented by proposing new models or checking predictive adequacy.

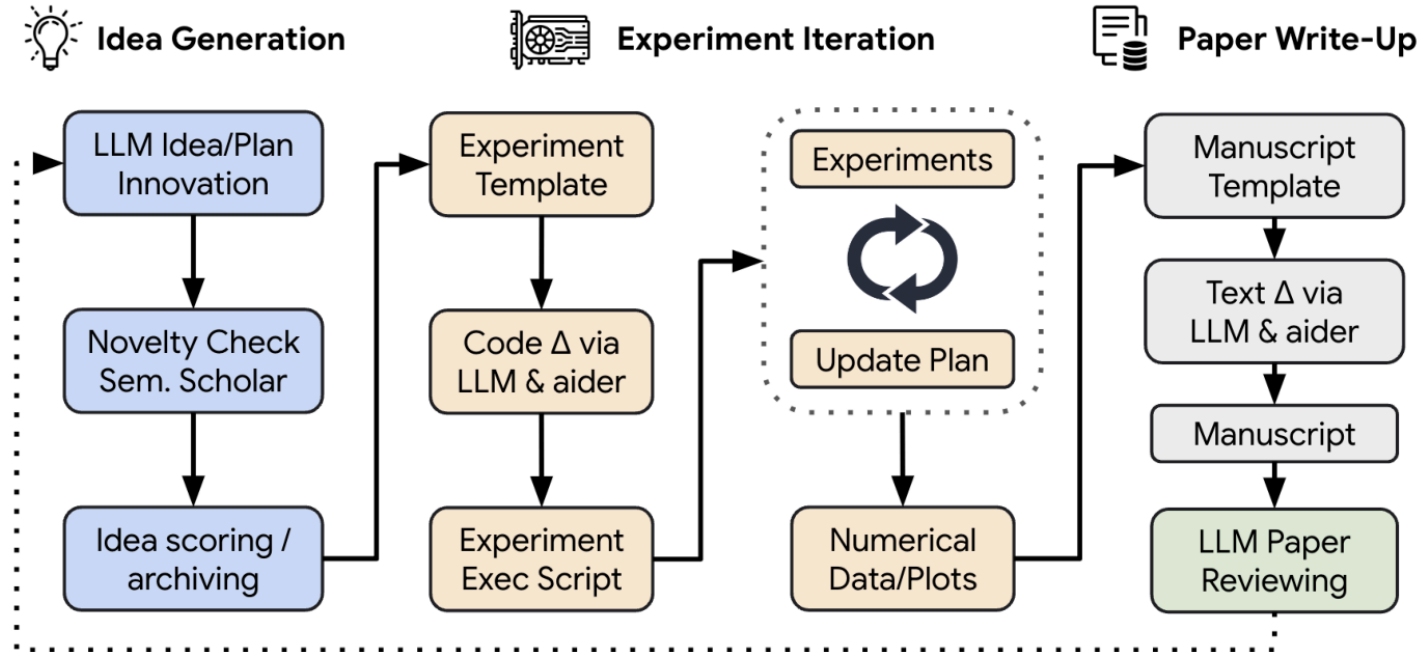
The AI Scientist: the method

An LLM proposes research ideas from a supplied experimental setting, checks nearby literature, modifies code, runs experiments, and assembles a manuscript.

Numerical outputs and plots inform subsequent iterations. An automated reviewer provides an additional assessment of the resulting paper.

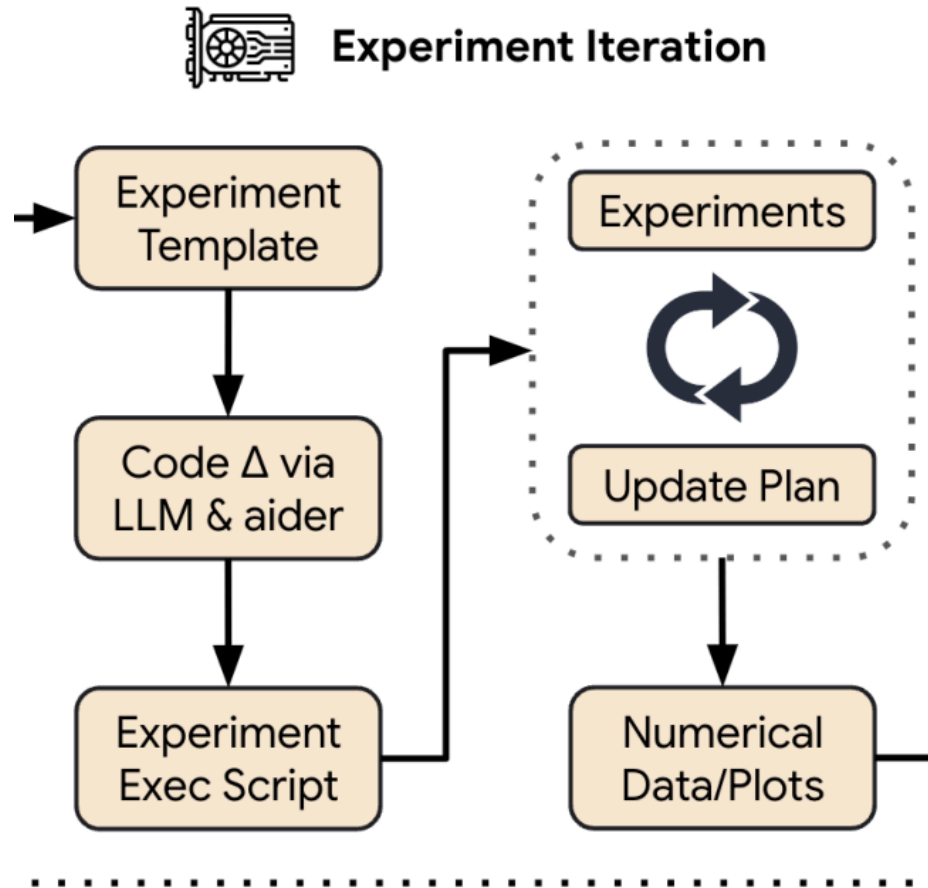
The workflow combines scientific assistance with orchestration and execution; the quality of each stage constrains the usefulness of the complete output.

The AI Scientist: research workflow



The three columns connect idea generation, experiment iteration and writing. Measurements and generated text travel through different parts of this loop. The novelty check and paper reviewer judge different targets; neither is an independent experimental replication.

The AI Scientist: mechanism detail



In the experiment loop, code changes produce numerical data and plots that feed an updated plan. Execution records distinguish changes supported by new observations from revisions to the written report.

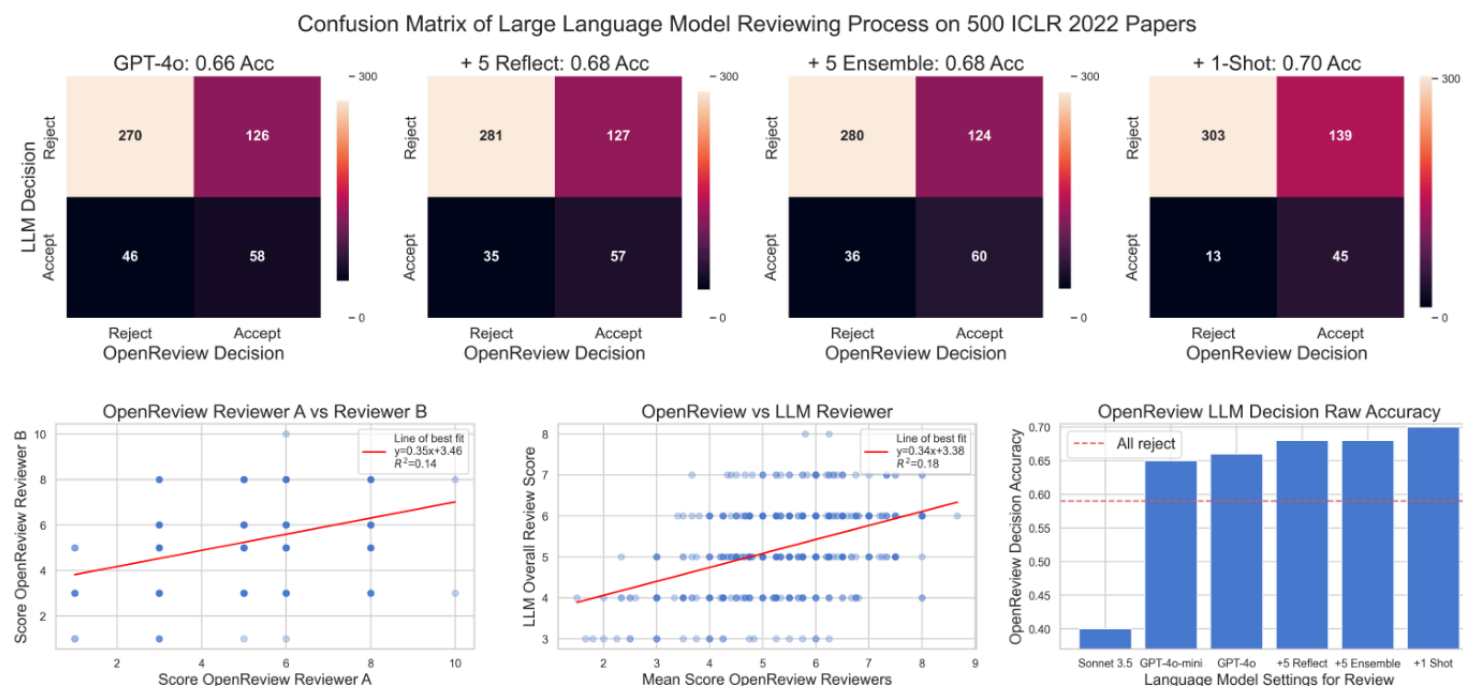
Evaluation scope

The paper evaluates its reviewing system against human conference decisions and demonstrates generated machine-learning research papers.

The reviewer comparison measures agreement with a reviewing reference; it does not directly measure whether a proposed scientific claim is true.

The labeled conditions vary reviewing prompts, ensembles and scoring. Those changes have a different target from the quality of the experiments being reviewed.

The AI Scientist: interpretation and limits



The experimental domains and templates restrict the action space. Novelty screening and automated review remain fallible, and successful code execution does not validate every inference in a manuscript.

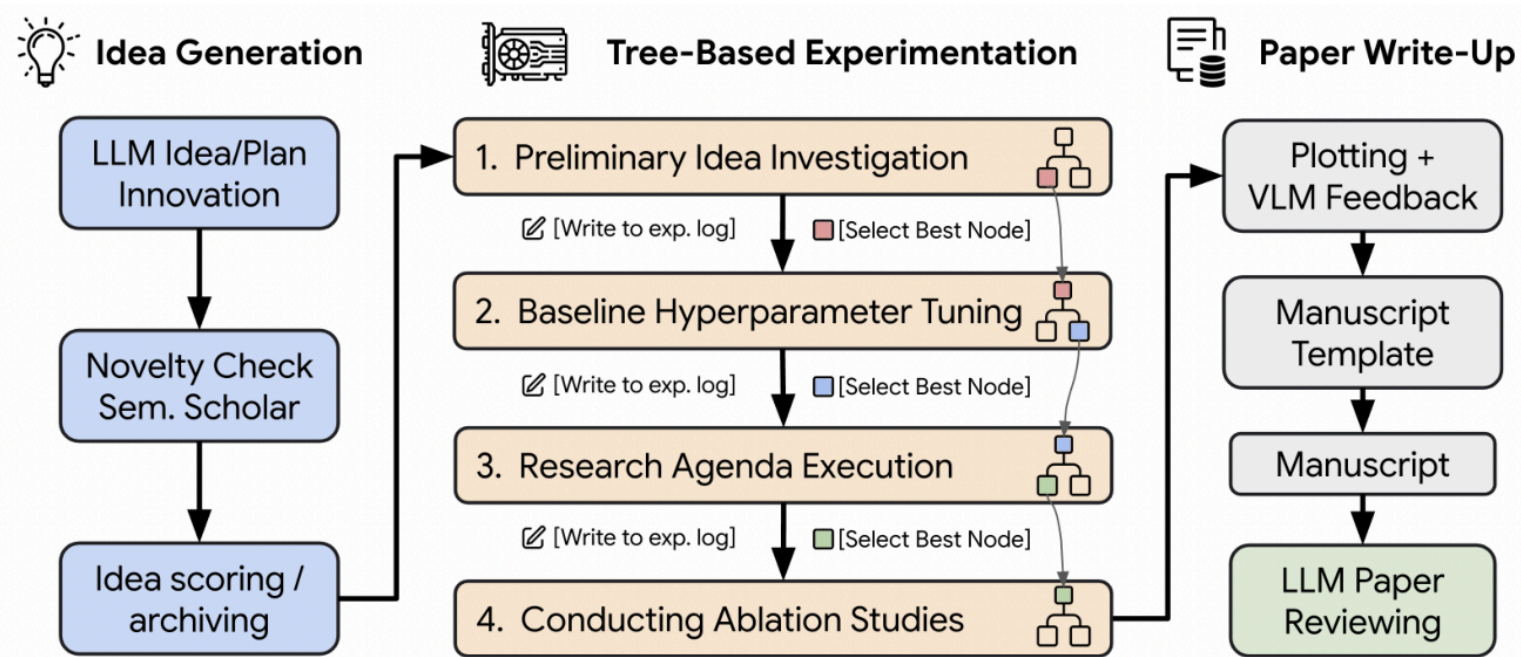
The AI Scientist-v2: the method

The workflow uses agentic tree search to organise experimental research rather than relying exclusively on a fixed experiment template.

Research stages include preliminary investigation, tuning, agenda execution, and ablation, followed by figure creation and writing. Candidate nodes preserve alternative computational paths.

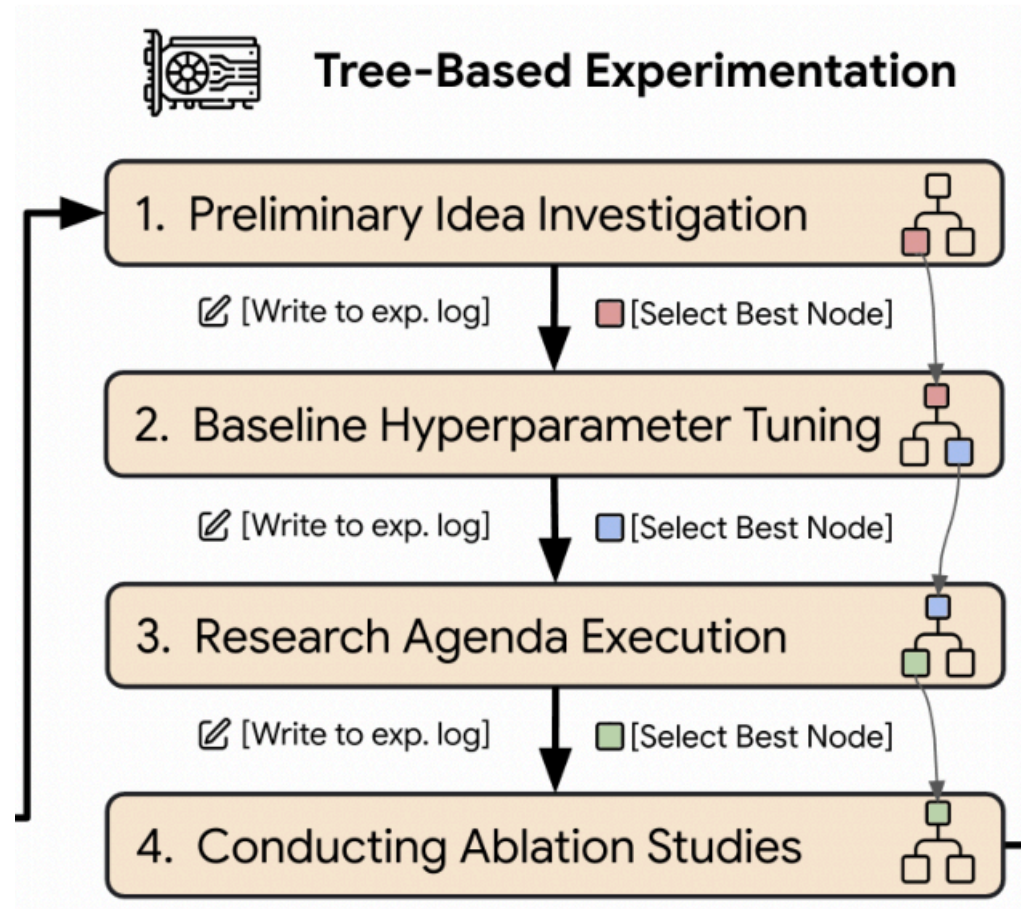
An evaluator helps choose promising branches, but the scientific interpretation of a branch still depends on its implementation and empirical evidence.

The AI Scientist-v2: staged search workflow



The four experimental stages retain selected nodes and pass their records to writing. The tree organizes alternatives and attempts. Scientific likelihood and novelty require evidence beyond the search structure.

The AI Scientist-v2: mechanism detail



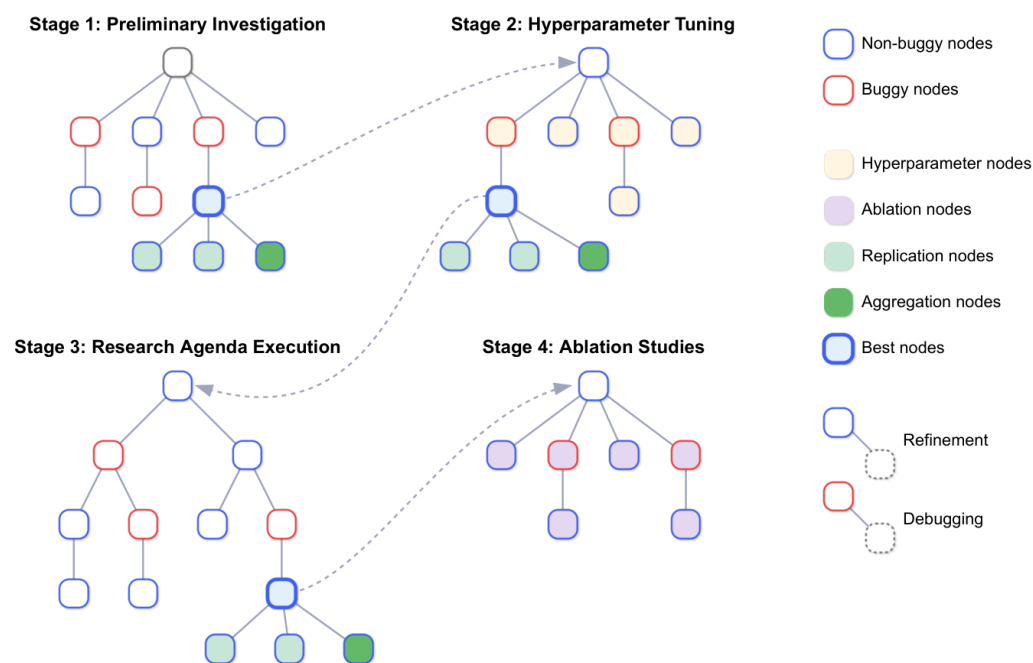
The transition from investigation to tuning selects a retained node. Its evidential basis and the treatment of discarded branches determine whether the resulting search trace supports the selection.

Evidence: a runtime allowance is not measured compute

Hyperparameter	Value
Debug Probability	1.0
Maximum Debug Depth	3
Maximum Experiment Runtime per Node	1 hour
<i>Node Allocation per Stage:</i>	
Stage 1: Preliminary Investigation	21 nodes
Stage 2: Hyperparameter Tuning	12 nodes
Stage 3: Research Agenda Execution	12 nodes
Stage 4: Ablation Studies	12 nodes

The AI Scientist-v2 configuration allocates 21, 12, 12, and 12 nodes across four stages, with a maximum experiment runtime of one hour per node. These are allowances, not observed consumption. The source separately describes a 15-hour overall runtime limit. Parallel execution and early completion make node-hours different from elapsed hours.

The AI Scientist-v2: interpretation and limits



Buggy and non-buggy branches lead to a selected checkpoint for the next stage. Debugging success and favorable model evaluation concern different targets. The workflow describes search organization rather than establishing novelty.

Revision can change the model class

A result can lower a candidate's probability, restrict its scope, split it into alternatives, or motivate a new mechanism.

These are useful record-level operations, not a substitute for scientific reasoning about why the change is warranted. Preserve previous versions and the evidence motivating each revision.

A model can legitimately retain its form after a noisy outcome; demanding a visible structural change after every measurement would be another error.

Stopping compares continuation with alternatives

Continuing can provide information, a better design, or a more reliable decision, but it consumes time and resources.

Compare expected future utility with execution costs and the value of reporting uncertainty or changing the question. A fixed budget is a legitimate stopping constraint when made explicit.

Information-per-cost thresholds are heuristics whose appropriateness depends on the horizon, objective, and possible complementarities between actions.

A closed-loop claim has several levels of autonomy

A system can propose actions, schedule computation, execute a constrained procedure, analyse results, or decide the next scientific question. Different studies automate different subsets.

State which decisions were made by the system and which depended on humans or fixed templates.

This permits a fair comparison of practical contributions without assuming that an experimentally useful system must own every stage of scientific work.

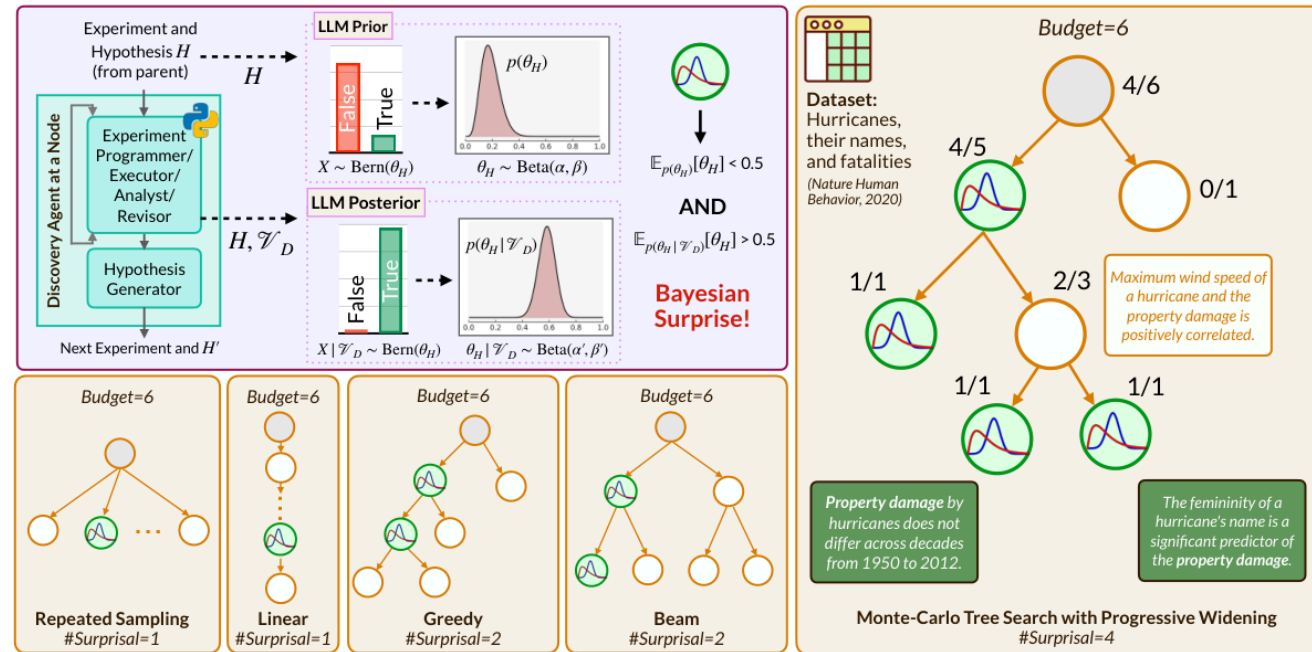
AutoDiscovery: the method

AutoDiscovery explores nested hypotheses using a search procedure guided by changes between elicited prior and posterior beliefs.

Data analysis and hypothesis evaluation produce feedback used in an MCTS-style exploration process. Bayesian surprise supplies an objective related to belief change.

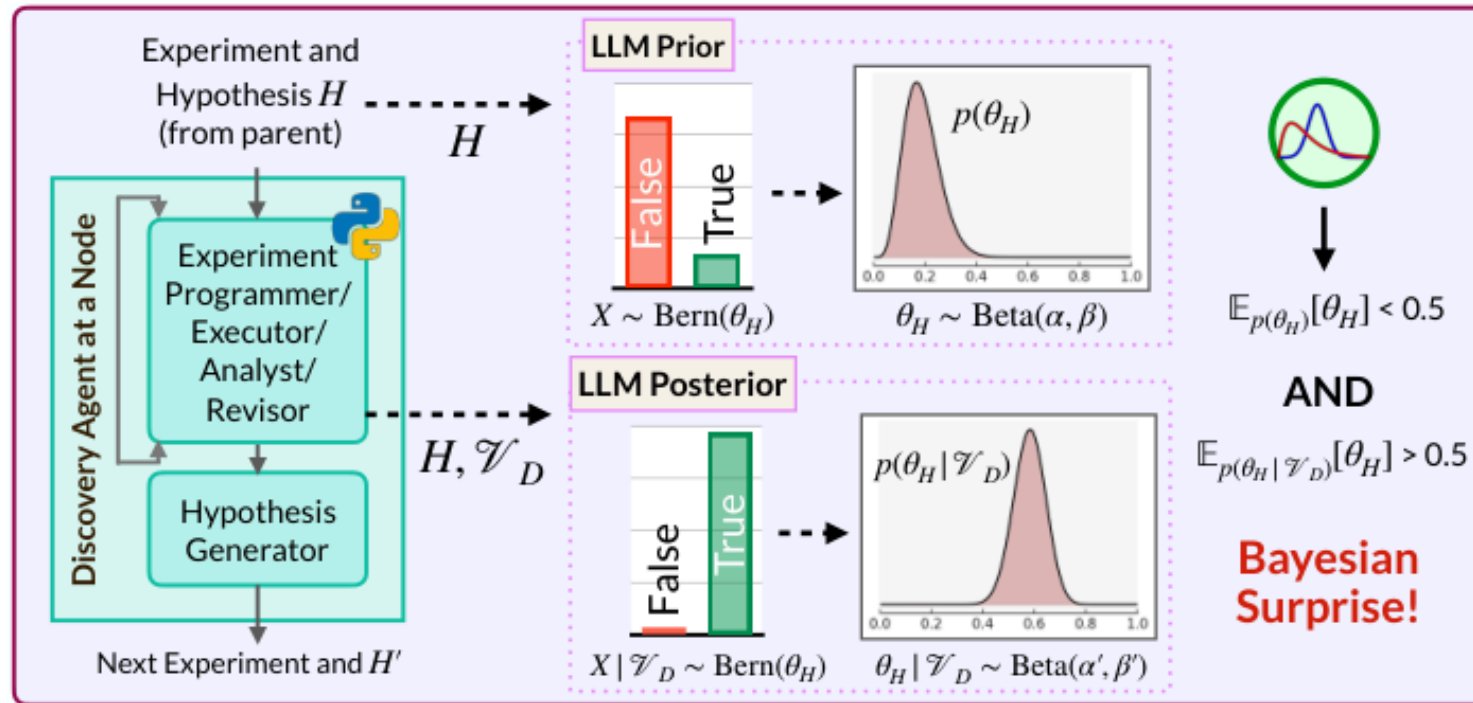
The method's operational belief estimates and search reward should be distinguished from a fully calibrated generative model and from expected information gain evaluated before an experiment.

AutoDiscovery: surprise-guided search



The figure links a hypothesis, analysis program, prior and posterior beliefs, and search expansion. Realized surprise concerns an observed outcome. Prospective information gain averages over outcomes that have not yet been observed.

AutoDiscovery: mechanism detail



The diagram links prior-to-posterior change to branch expansion. Whether a search reward is available before or after data analysis determines its decision-theoretic interpretation.

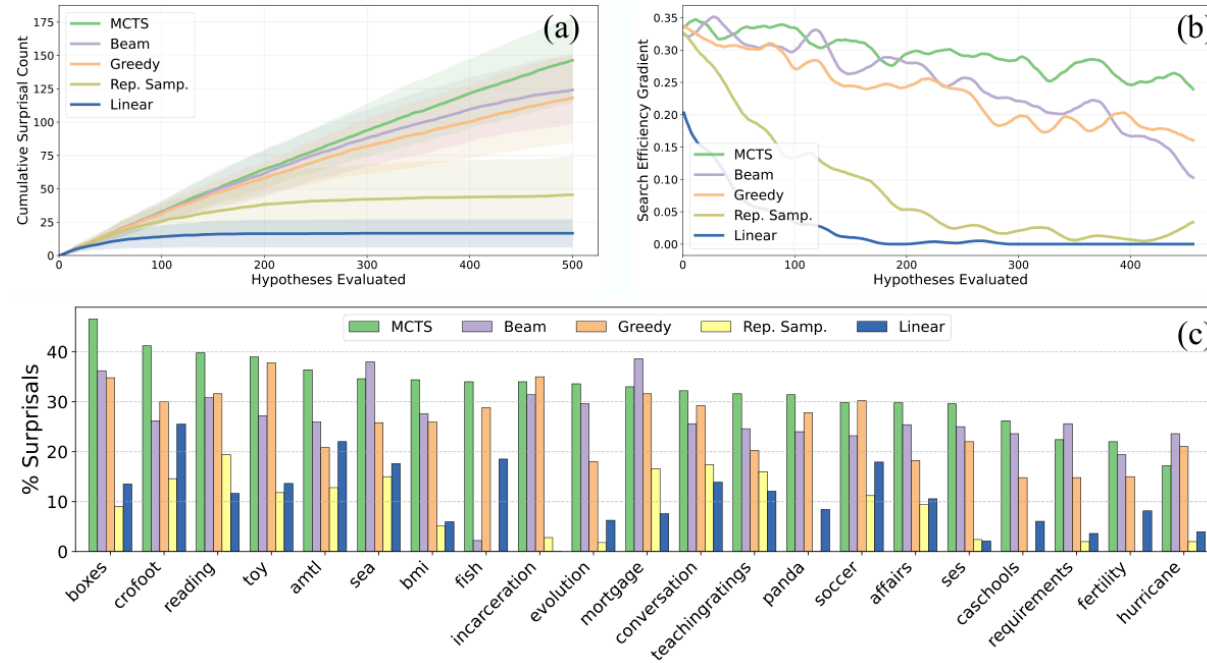
Evaluation scope

The study compares search procedures using discovery-related rewards and human assessments of selected outputs.

The original search-performance figure separates cumulative discoveries, yield, and performance across datasets. Interpret each axis with its evaluation definition.

A high surprise score can help prioritise investigations, but its scientific value depends on the reliability of the analysis and the relevance of the target belief.

AutoDiscovery: interpretation and limits



Elicited beliefs can be miscalibrated, and an unexpected result can reflect a model or data error. Human assessments and benchmark-specific rewards do not imply a universal stopping or experiment-selection rule.

An execution failure and a null result update different things

Feedback	What is justified?	Update target
Import error before any measurement	No scientific result was produced	Repair the environment or tool call
Fresh valid test finds no residual effect	A null can distinguish explanations	Update the scientific model weights
$P(\text{null} \mid A) = 0.9$; $P(\text{null} \mid B) = 0.2$	Starting from $P(A) = 0.8$	$P(A \mid \text{null}) = 0.72/0.76 \approx 0.947$
The valid result is copied into another report	It is the same evidence again	Do not apply the likelihood ratio twice

Synthetic continuation of the sensor case. The null likelihoods and conditional independence are stipulated; a tool crash has no such likelihood.

GEPA revises the failing component

Execution evidence	Reflection	Candidate change
Second-hop query repeats the question	The first result already resolves part of the task	Ask for the missing entity or relation
Retrieved passages miss the needed broader entity	The query is too narrow	Broaden the next query using first-hop context
A revised prompt succeeds on a minibatch	The improvement may be local	Evaluate on the larger development set
Candidates win on different cases	One average can hide complementarity	Retain useful candidates on a Pareto frontier

GEPA (ICLR 2026): reflective prompt evolution with fixed model weights. The query example paraphrases its published Figure 2.

TextGrad sends criticism to an editable variable

Forward object	Feedback path	Revision target
Instruction → generated answer	An evaluator identifies a specific failure	The instruction text
Answer cites an irrelevant observation	Feedback describes the missing evidence relation	Require source-supported intermediate claims
Revised instruction is executed	Measure the resulting answer, not its wording	Keep or reject the proposed edit
The evaluation itself is wrong	Its criticism can send revision in the wrong direction	Inspect evaluator reliability separately

TextGrad propagates textual feedback through a computation graph. “Gradient” is an analogy for directed feedback, not an ordinary derivative.

A Pareto archive preserves complementary successes

Candidate	Three development cases	Retention decision
A	[1, 1, 0]	Keep: covers cases 1 and 2
B	[0, 0, 1]	Keep: uniquely covers case 3
C	[1, 0, 0]	Dominated by A on these cases
Final procedure chosen	Freeze prompts, selection and evaluation rules	Then measure on untouched confirmation cases

Synthetic vectors illustrate dominance, not GEPA results. B survives despite a lower mean because its contribution is complementary.

Return to the two executable hypotheses

L03 and L12: M1 has $X=U$, $Y=X$; M2 has $X=U$, $Y=U$; U is a fair bit. Observational agreement left these models tied.

Now set $X=1$ in fresh units. Add a stated measurement model: readout O flips the true binary Y with probability 0.1, independently across units.

Then $P(O=1 \mid M1)=0.9$, while $P(O=1 \mid M2)=0.5$. These are predictive probabilities from the models, not a critic's confidence scores.

Four results turn predictions into posterior odds

Begin with equal prior probabilities for M1 and M2. One positive readout has likelihood ratio $0.9/0.5 = 1.8$.

For four independent positive readouts, posterior odds are $1 \times 1.8^4 = 10.4976$. Thus $P(M1 \mid \text{data}) = 10.4976/11.4976 \approx 0.913$.

The support is relative to this candidate pair and error model. More confident selection is not proof that M1 describes the whole system.

A contrary result changes the next decision

The fifth readout is $O=0$. Its likelihood ratio is $0.1/0.5 = 0.2$, so the odds fall to $10.4976 \times 0.2 = 2.09952$.

The revised probability of M1 is $2.09952/3.09952 \approx 0.677$. Evidence need not erase earlier support to make the conclusion less secure.

A repeat may distinguish the models further; a measurement check may test the error assumption. Their relative value depends on costs and the decision at stake.

Revision can target the hypothesis or the evidence model

What was learned?	Appropriate revision	What would be a mistake?
A fresh measured outcome	Update model odds using its conditional likelihood.	Treat a model-generated comment as new measurement.
Shared calibration error	Infer a common error term; recompute dependence.	Multiply likelihood ratios as if errors were independent.
Both models fit poorly	Construct and test a new mechanism or noise model.	Call the less-bad model true because its relative odds rise.

A closed loop revises representations and measurement assumptions as well as ranking existing candidates.

Pedagogical reconstruction

[Pearl \(2009\)](#)

A loop evaluation should include stopping behaviour

Simulated campaigns with useful, redundant, misleading and infeasible actions expose different closed-loop failure modes. Equal resources and observation delays support controlled system comparisons.

Final decision utility, informative actions, unnecessary repetitions, invalid updates and residual uncertainty describe different consequences of a closed loop.

Some tasks warrant stopping or changing the model class. Completed-cycle counts alone reward persistence even when further work has no scientific value.

The same update expressed in log-odds

Let ℓ be the logarithm of odds for M1 over M2. Each new readout adds $\ln[P(O \mid M1, \text{history}) / P(O \mid M2, \text{history})]$.

Here the prior log-odds are 0. Four positives add $4 \ln 1.8 \approx 2.351$; one negative adds $\ln 0.2 \approx -1.609$, leaving $\ell \approx 0.742$.

Convert back: $P(M1 \mid \text{data}) = 1/(1 + \exp(-\ell)) \approx 0.677$. The concise formula is valid only with the likelihood assumptions made explicit.

Testing whether observations affect later actions

With the initial question, candidate set and random seed fixed, observations with different implications reveal whether the subsequent action depends on the evidence.

An unchanged policy can reflect an ignored observation, an uninformative test, or two different posteriors that favor the same action.

A useful feedback test needs a case where the decision model predicts different actions. Different wording alone is not evidence of adaptive experimentation.

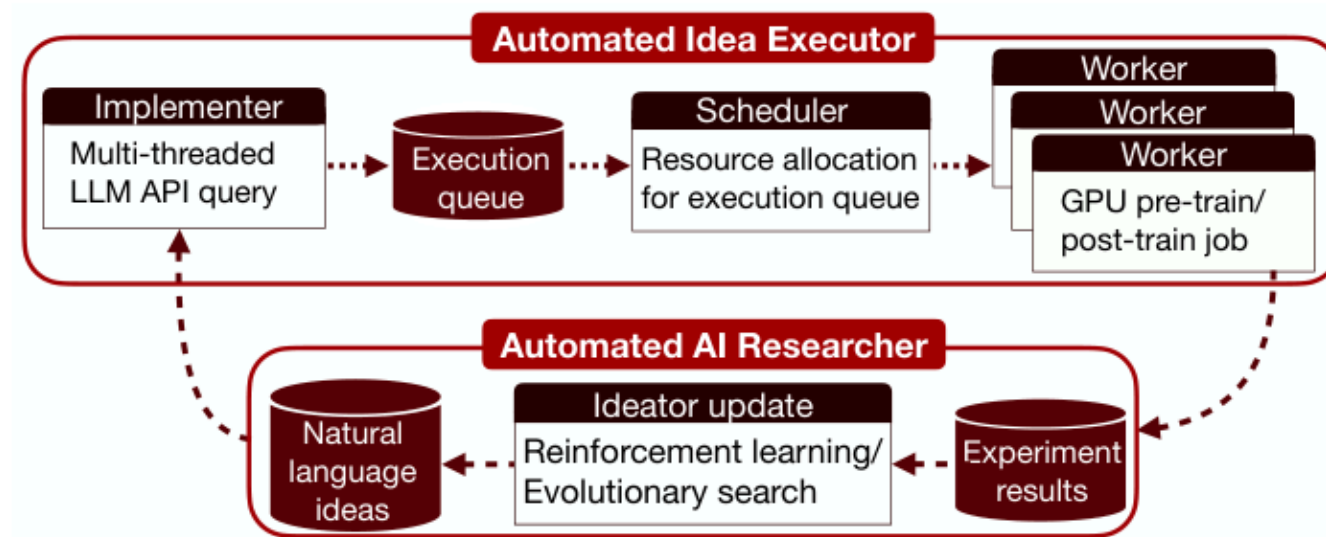
Recent research: Execution-grounded AI research

The study separates an ideator from an automated executor containing implementation, scheduling, and worker components.

Proposed ideas are implemented and tested in defined machine-learning environments. Execution feedback can then guide evolutionary search or reinforcement learning of the ideator.

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.

Execution-grounded research: feedback architecture



The architecture links a natural-language proposal to implementation, an execution queue and worker results, then returns feedback to the researcher. It explicitly separates proposing an idea from obtaining an executed result.

Recent methods in context

Model Discovery Agent (preprint; revised 2026). Combines Bayesian model/parameter inference, LLM proposals of model structures and value-of-information experiment choice. Version 4 reports fixing prompt leakage and rerunning benchmarks; comparisons must use the corrected version.

GEPA (ICLR 2026). GEPA adapts prompts using reflected execution feedback and a population of candidates. It changes instructions rather than model weights, and its development feedback remains separate from final evaluation.

CHIVE (Aug 2026 preprint). Investigates observed model behaviors with counterfactual prompt edits and tests explanations through predictive usefulness. The reported lack of uplift from tested interpretability tools is conditional on this evaluation and tool set.

References (1/2)

[Chris Lu, Cong Lu, Robert Tjarko Lange et al. \(2024\). The AI Scientist: Towards Fully Automated Open-Ended Scientific Discovery](#)

[Yutaro Yamada, Robert Tjarko Lange, Cong Lu et al. \(2025\). The AI Scientist-v2: Workshop-Level Automated Scientific Discovery via Agentic Tree Search](#)

[Dhruv Agarwal, Bodhisattwa Prasad Majumder, Reece Adamson et al. \(2025\). AutoDiscovery: Open-ended Scientific Discovery via Bayesian Surprise](#)

[Samuel Schmidgall, Xiaokai Zhu, Marian Shaw et al. \(2026\). Accelerating Scientific Research with Gemini in the Real-World](#)

[Chenglei Si, Zitong Yang, Yejin Choi et al. \(2026\). Towards Execution-Grounded Automated AI Research](#)

[Judea Pearl \(2009\). Causal inference in statistics: An overview](#)

References (2/2)

[Lakshya A Agrawal, Shangyin Tan, Dilara Soylu et al. \(2025\). GEPA: Reflective Prompt Evolution Can Outperform Reinforcement Learning](#)

[Mert Yuksekgonul, Federico Bianchi, Joseph Boen et al. \(2024\). TextGrad: Automatic "Differentiation" via Text](#)

[Kevin Murphy \(2026\). Model Discovery Agent: LLM-assisted Bayesian experiment design for data-efficient discovery of mechanistic world models](#)

[Adam Karvonen, Euan Ong, Subhash Kantamneni et al. \(2026\). Would this change your answer? Evaluating Explanations of LLM Behavior In The Wild with Counterfactual Experiments](#)

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