Review

Not peer-reviewed version

Closing the Loop in AI-Driven Biomedical Discovery

Ada Fang , Kevin Li †, Ayush Noori †, Lukas Fesser †, Marinka Zitnik *

Posted Date: 28 August 2026

doi: 10.20944/preprints202608.2107.v1

Keywords: AI scientists; autonomous scientific discovery; closed-loop discovery; biomedical discovery; agentic AI; scientific reasoning; experimental verification; reinforcement learning

 doi:10.20944/preprints202608.2107.v1

Review

Closing the Loop in AI-Driven Biomedical Discovery

Ada Fang [1,2,3], Kevin Li [1,4,†], Ayush Noori [1,5,†], Lukas Fesser [1,3,6,†] and Marinka Zitnik [1,3,7,8,*]

[1] Department of Biomedical Informatics, Harvard Medical School, Boston, MA, USA
[2] Department of Chemistry and Chemical Biology, Harvard University, Cambridge, MA, USA
[3] Kempner Institute for the Study of Natural and Artificial Intelligence at Harvard University, Allston, MA, USA
[4] Harvard-MIT Program in Health Sciences and Technology, Massachusetts Institute of Technology, Cambridge, MA, USA
[5] Department of Engineering Science, University of Oxford, Oxford, UK
[6] Applied Mathematics Program, Harvard John A. Paulson School of Engineering and Applied Sciences, Allston, MA, USA
[7] Broad Institute of MIT and Harvard, Cambridge, MA, USA
[8] Harvard Data Science Initiative, Cambridge, MA, USA
* Correspondence: marinka@hms.harvard.edu
† These authors contributed equally as co-second authors

Abstract

AI scientists generate hypotheses, propose experiments, and analyze datasets, and several have produced findings that were confirmed in the laboratory. Although they are opening a path to autonomous discovery, the loop from hypothesis to experiment to revised hypothesis is still often anecdotal. Here, we consider how that loop could be closed to enable AI-driven biomedical discovery. Closing the loop requires AI scientists that reason over biomedical knowledge and data across horizons of days to months. It also requires verifying hypotheses with computational tools and experiments in biological laboratories and clinical environments. Agentic and reasoning methods need to learn over time from sparse, delayed, and noisy feedback that verifiers return. In mathematics and program search, where candidate solutions can be verified at negligible cost and often as soon as they are generated, methods that generate and score large numbers of candidates have improved bounds on problems that had been open for decades. In biomedical sciences, comparable progress requires closing the loop where each test is slow, costly, and partially informative. Closed-loop AI-driven biomedical discovery can enable a mode of research in which AI continuously reviews hypotheses against accumulating experimental results. Scientists choose which experiments to pursue, and AI maintains a record that links each result to the hypothesis that produced it.

Keywords: AI scientists; autonomous scientific discovery; closed-loop discovery; biomedical discovery; agentic AI; scientific reasoning; experimental verification; reinforcement learning

Introduction

Autonomous discovery is a mode of research in which AI takes on hypothesis generation, experimental design, and data interpretation alongside scientists directing the work (Figure 1). Systems that operate in this way are increasingly referred to as AI scientists: agentic systems that proceed through many steps of reasoning interleaved with analysis of multimodal data to complete a research objective. Closed-loop AI scientists go further by running research cycles that proceed through generating hypotheses, selecting experiments or computational tools to test them, interpreting the resulting evidence, and using that evidence to determine subsequent actions [1,2]. Recent AI scientists have advanced from executing fixed computational workflows to performing complex, open-ended analyses. ChemCrow, for example, answers chemistry questions by calling tools rather than by recalling facts from LLM parameters [3], data-to-paper takes a dataset through analysis to a manuscript whose claims are traceable to the code that produced them [4], and Kosmos generates and ranks mechanistic hypotheses that are then sent to a laboratory to test them [5]. AI-generated scientific outputs have also undergone experimental validation, with laboratory results supporting several of them. Coscientist wrote and

 doi:10.20944/preprints202608.2107.v1

executed protocols for palladium-catalyzed cross-coupling reactions on a robotic platform [6], and the A-Lab, a closed-loop autonomous laboratory, produced previously unreported inorganic compounds within weeks [7]. In biomedicine, Google's Co-Scientist proposed mechanistic accounts of antimicrobial resistance and candidate treatments for liver fibrosis that collaborating laboratories confirmed experimentally [8], Robin generated candidate therapeutics that its authors evaluated in cell-based assays [9], and PROTON generated neurological hypotheses that molecular, organoid, and clinical experiments then supported [10].

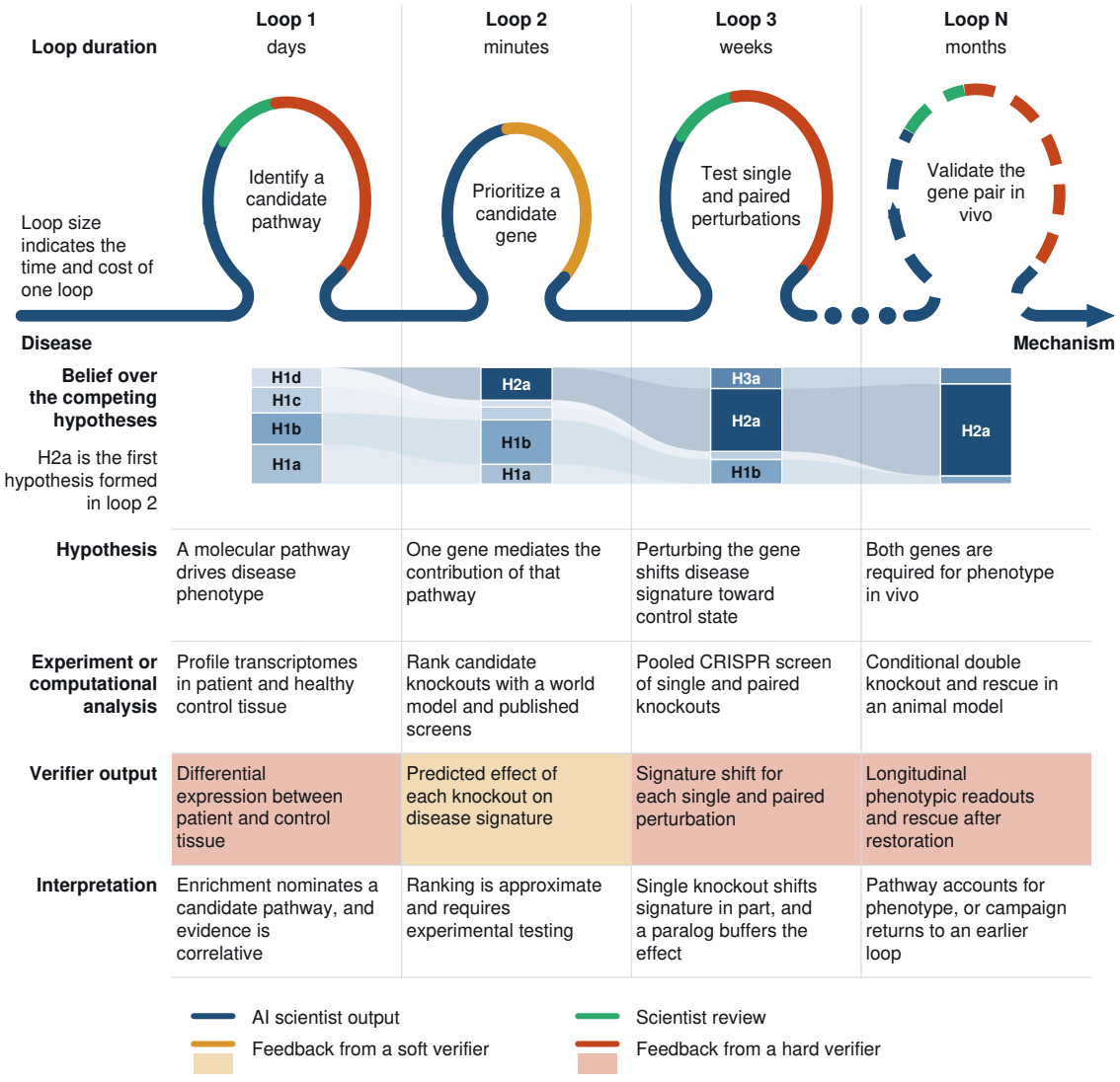

Figure 1. Closing the loop in AI-driven biomedical discovery. A campaign advances from a disease observation to a validated mechanism through repeated loops. In each loop, the AI scientist forms or revises a hypothesis, selects an experiment or computational query to test it, and interprets the resulting evidence. The verifier returns a computational result or experimental measurement, from which evidence is derived and used to update the belief over competing hypotheses. Loop size indicates the time and cost of one loop, and the colored segment indicates the type of verifier queried and the time required to obtain its output. A scientist reviews actions whose cost or risk warrants oversight. As evidence accumulates, beliefs are revised, hypotheses may be discarded or newly formed, and the hypothesis favored early in the campaign need not be the one ultimately supported.

Fields such as mathematics and program search have closed the loop, since AI scientists can generate many candidate solutions and test them with verifiers that evaluate correctness exactly, typically at negligible cost. For example, a proof assistant checks a candidate proof and a coding agent can execute a candidate program against a suite of test cases, at a cost far below that of generating it.

 doi:10.20944/preprints202608.2107.v1

Models can exploit this by generating and scoring a large number of candidates, which is how formal theorem proving solved Olympiad problems in mathematics [11–13], how reasoning models trained against verifiable rewards acquired their capabilities [14,15], and how evolutionary program search improved bounds on long-open combinatorial problems and produced faster kernels that now run in production [16,17]. Closing the loop in these domains depends on the reliability of verifiers. A proof checker or a test suite returns an exact answer, whereas learned judge models evaluate candidate solutions approximately.

In biomedical discovery, however, establishing the mechanism behind a hypothesis generated by an AI scientist requires evidence from several verifiers, and the result from any single verifier captures only one state of the relevant biological system. An AI scientist needs to reconstruct the full state across loops of molecular, cellular, and clinical experiments, each of which reports on a different aspect of the same mechanism [18–21] (Figure 1). In a single loop, an AI scientist can generate thousands of candidates, of which a laboratory tests only a handful, meaning that the allocation of budget to hypothesis verifiers restricts discovery as much as the quality of generated hypotheses [22]. A perturbation assay can require days of execution and return a noisy readout from one point in a large intervention space, and clinical validation can extend over years at a cost that constrains how many hypotheses a campaign can test [6,10]. A campaign can also run for weeks to months, and a negative result obtained in an early loop can remain the constraint that excludes a hypothesis considered much later, after the reasoning that produced it has been superseded. An AI scientist operating under these conditions has to determine which hypotheses warrant evaluation in a laboratory and which verifier to query and at what fidelity and stage of each loop.

In this Review, we examine autonomous AI scientists that could close this loop in biomedical discovery, and we treat verification as the problem on which closing the loop depends. Each loop of a campaign is closed when a verifier returns a computational result or experimental measurement to test the hypotheses produced by the AI scientist. The AI scientist uses this evidence to update its beliefs, which in turn guide the actions selected in the next loop (Figure 1). Verifiers can be computational or experimental, and the AI scientist must decide which source to query for each output and whether the returned evidence is sufficient to act. Over a campaign that may run for weeks or months, verification repeatedly determines when the AI scientist can update its hypotheses and whether it has enough evidence to proceed. Autonomous biomedical discovery will advance as verification matches the rate at which AI scientists generate hypotheses so that a campaign carries each hypothesis to the experiment that tests it and back to a revised hypothesis.

Overview of Closed-Loop AI Systems

A closed-loop AI scientist (Box 1) operates autonomously on an open-ended scientific question, one for which the set of candidate explanations is not specified in advance. In each loop it formulates hypotheses, designs and executes experiments, analyzes the resulting measurements to obtain evidence, interprets that evidence, and updates its belief over hypotheses to develop an increasingly explanatory and predictive account of the biological system under study.

Each loop produces a hypothesis, the experiment or computational query selected to test it, the protocol that implements that selection and, once the verifier output is available, an interpretation of that evidence and an updated belief over the competing hypotheses that conditions the next loop. We refer to these collectively as AI outputs. Computational results and experimental measurements returned by verifiers are the external inputs to the loop, and the evidence used for belief revision is derived from those outputs. Here we outline the components that a closed-loop AI scientist requires and the contribution each makes to the loop.

Reasoning models. The reasoning model generates scientific inferences and candidate actions from the history accumulated across loops. Together with the harness, it determines the next reasoning or experimental action. It maintains a representation of that history in its context and in external memory [23], including available evidence, competing hypotheses, uncertainty, prior actions, and

 doi:10.20944/preprints202608.2107.v1

experimental constraints. It integrates heterogeneous sources of information, generates and revises testable hypotheses, derives predictions, and selects the next action expected to be most informative. It must also interpret positive, null, and failed experimental results to adjust the posterior probability of its hypotheses. Reasoning models are at present implemented as LLMs. An LLM inherits the distribution of the corpus it was pretrained on, and because the published literature overrepresents positive findings, an LLM trained on it acquires the same disposition [24–26].

Computational actions. These actions acquire or generate evidence *in silico* before the AI scientist commits resources to physical experiments. They include querying scientific databases and searching the literature; running mechanistic or data-driven simulations; evaluating surrogate predictive models; querying a world model, which represents a biological system closely enough to predict how it responds to an intervention before that intervention is performed [27,28]; estimating counterfactual outcomes; and comparing candidate experimental designs. A computational action functions as a soft verifier when it assesses an AI output specified before the query.

Experimental actions. Experimental actions physically intervene on the biological system under study to generate new measurements. They include selecting perturbations, biological contexts, measurements, controls, doses, and time points, as well as executing experiments through self-driving laboratories or human collaborators. Effective action selection should account not only for the probability of obtaining a desired result but also for an experiment's ability to discriminate among competing hypotheses, reduce uncertainty, and advance the scientific objective under cost, time, and safety constraints. Because laboratory throughput limits how many candidates can be tested, AI scientists can evaluate candidates in batches using multiplexed screens and pooled perturbation assays [29,30]. Whether conducted individually or in a batch, an experiment serves as an experimental verifier by generating measurements from a prospectively specified test of a hypothesis or prediction.

Verifier outputs and evidence. A verifier returns either a computational result or an experimental measurement from a prospectively specified test of a hypothesis, prediction, or proposed action. Computational and experimental verifiers differ in their cost, latency, reliability, and how directly they interrogate the biological system. Their outputs are analyzed to determine what evidence they provide for or against the competing hypotheses. The AI scientist selects which verifier to query by weighing these properties and its expected informativeness, then updates its belief over hypotheses according to the resulting evidence. That revised belief conditions the action selected in the next loop.

AI scientist harness. The agentic harness includes skills and prompts that determine how actions are orchestrated and how the history is updated [31,32]. Importantly, the harness provides guidance on the policy over actions based on the current history. For example, it can encourage actions with little immediate return that can still be worth taking, such as establishing a cell line or collecting the measurements that calibrate a surrogate model to prioritize which experiments later loops run. Here maximizing the expected gain at the current loop alone forgoes the value that accrues over a campaign. A stochastic policy can also serve a campaign better than a deterministic one, since sampling among comparably informative experiments keeps several mechanistic accounts under test and guards against diversity collapse, in which the hypotheses a loop generates narrow to the band its current criterion rewards [33]. Sequential Bayesian experimental design supplies criteria for choosing among candidate actions under a budget [34–36].

Improving AI scientists. This can be achieved by building better components, such as reasoning models and world models, or by better orchestration of those components, which are improvements to the agentic harness [33]. AI scientists can also evolve at deployment time and improve their own harness [37]. Determining where AI scientists can be improved requires inspecting the discovery trajectory and the sequence of intermediate steps connecting the measurements consulted to the hypothesis proposed/the experiment selected. These discovery trajectories serve a second role as training data, which we take up in the next section.

 doi:10.20944/preprints202608.2107.v1

Human and AI co-science. Current AI scientists operate under human-in-the-loop oversight [2,38–41], and a scientist reviews each transition before the AI scientist proceeds. Scientists select which hypothesis to pursue, design the assay, judge whether a readout supports or refutes the hypothesis, and decide what to measure next. Closing the loop moves parts of a campaign to human-on-the-loop discovery [42,43], in which the AI scientist carries the campaign from one transition to the next while scientists audit AI outputs and gate the actions that warrant review. Such an operational model is possible when the action space is fixed, so that the campaign tests hypotheses over a small set of predefined variables rather than deciding which question to investigate. Oversight then scales with the cost and risk of the action. A retrieval query or a computational model evaluation can proceed without review, whereas an experiment that consumes scarce material or raises a biosafety concern cannot proceed until a scientist approves it [38].

Box 1 | Key definitions

Trace. A reasoning trace is a sequence of tokens consisting of explicit intermediate reasoning steps followed by a final decision, answer, or conclusion. We define a reasoning trace ρ as

$$\rho = (k_1, k_2, \ldots, k_N, o_1, o_2, \ldots, o_L),$$

where $\{k_n\}_{n=1}^{N}$ are reasoning tokens corresponding to intermediate reasoning logic and $\{o_l\}_{l=1}^{L}$ are final output tokens describing the final conclusion.

Trajectory. Let $h_t = (s_{0:t}, a_{0:t-1})$ denote the *history* available at step t, where s_t denotes the observations received at step t and a_t denotes the action selected at step t by the AI scientist, and ρ_t denotes the reasoning trace associated with that action. Observations may span modalities $s_t = \{x_t^{(m)} : m \in \mathcal{M}_t\}$, such as text, sequences, molecular structures, or perturbation responses, from which the AI scientist infers the latent biological state z_t. It separately maintains a set of competing hypotheses $\mathcal{H}_t$ and a *belief* b_t over those hypotheses, where

$$b_t(H) \geq 0, \qquad \sum_{H \in \mathcal{H}_t} b_t(H) = 1.$$

A new verifier observation s_{t+1} provides evidence that can revise the belief over existing hypotheses and motivate hypotheses to be generated, revised, or discarded, yielding

$$(\mathcal{H}_t, b_t) \xrightarrow{s_{t+1}} (\mathcal{H}_{t+1}, b_{t+1}).$$

We define a trajectory τ as

$$\tau = (s_0, a_0, s_1, \ldots, a_{T-1}, s_T).$$

Loop. One pass of the discovery process, in which the AI scientist forms or revises a hypothesis, selects and executes an action a_t, observes s_{t+1}, and updates its belief over the competing hypotheses. A loop closes when a verifier returns an observation from which evidence is derived for the tested hypothesis.

Campaign. A series of loops directed at one scientific objective. A campaign seeks to maximize scientific utility across the discovery trajectory while allocating a limited verification budget over the sequence of actions. Campaigns can run for weeks to months, so evidence from an early loop can constrain a hypothesis formed much later, and the verification budget must be allocated across the series rather than loop by loop.

AI scientist. An AI scientist comprises the reasoning model, the harness that orchestrates it, the computational and experimental actions available to it, and the verifiers that supply external

 doi:10.20944/preprints202608.2107.v1

observations from which evidence is derived. Together, the reasoning model and harness induce the AI scientist's policy over scientific actions,

$$\Pi(a_t \mid h_t).$$

Reasoning model. The reasoning model generates scientific inferences and candidate actions from the history h_t. At present, this component is instantiated by an LLM parameterized by θ, which defines a distribution π_θ over token sequences. We use *LLM* to refer to properties of the underlying model, such as its parameters and pretraining corpus, and *reasoning model* to refer to the role that model plays in scientific reasoning and action generation within the discovery loop. The reasoning model is optimized or evaluated over the resulting discovery trajectory, rather than only on the accuracy of a single prediction.

Harness. The harness is the scaffolding around the reasoning model. It comprises skills, prompts, and orchestration logic that determine which actions are available and how h_t is maintained. It is revised at deployment rather than by gradient updates to θ.

Soft verifier. A soft verifier evaluates an AI scientist's output computationally, returning the next observation

$$s_{t+1} \sim p_{\mathrm{s}}(s_{t+1} \mid a_t).$$

Predictive models and simulations can approximate experimental outcomes, whereas retrieval systems and learned judges assess consistency with existing evidence or specified criteria. These assessments provide scalable feedback, but they can be noisy, biased, poorly calibrated, or exploitable when used to construct rewards.

Hard verifier. A hard verifier queries the physical or prospective environment through an intervention a_t, producing the next observation

$$s_{t+1} \sim p_{\mathrm{env}}(s_{t+1} \mid \mathrm{do}(a_t), \kappa).$$

Here, κ denotes the experimental or clinical context. When a soft verifier serves as a surrogate for a physical experiment, p_{s} approximates p_{env}. Wet-lab assays, robotic experiments, and prospective studies provide higher-fidelity evidence than computational proxies, although their outcomes remain noisy, context-dependent, costly, and delayed.

World model. A world model represents the dynamics and observables of the biological system under study, for example through

$$p_\phi(z_{t+1} \mid z_t, \mathrm{do}(a_t)), \qquad p_\phi(s_{t+1} \mid z_{t+1}).$$

It supports prediction under interventions, counterfactual simulation, experiment selection, and belief updating.

Training Reasoning Models for Closed-Loop Discovery

The reasoning model determines which hypotheses an AI scientist considers, which experiments it selects, and how it interprets the resulting observations. These decisions require biomedical knowledge and depend on one another across a campaign, so an error in one loop can propagate to the hypotheses and actions selected in later loops. Training objectives must therefore account for decisions made within individual loops and their consequences across the discovery trajectory rather than only the accuracy of a single answer.

Training combines domain specialization with two principal post-training paradigms. Continual pretraining adapts the LLM to scientific corpora. Supervision over reasoning traces shapes the

 doi:10.20944/preprints202608.2107.v1

intermediate steps it generates [44,45], whereas reinforcement learning (RL) optimizes generated traces or action policies against constructed reward signals [14]. An LLM parameterized by θ defines a distribution π_θ over token sequences. Together with the harness, the reasoning model induces the AI scientist's policy $\Pi(a_t \mid h_t)$ over scientific actions. Reward design determines whether RL favors scientifically useful properties [8,36], including whether a proposed experiment is feasible and whether its outcome is expected to discriminate among the accounts still under consideration.

Continual pretraining (CPT) [46] specializes a general-purpose foundation model on scientific corpora such as literature, protocols, and databases (Figure 2a). For a sequence of text tokens $w = (w_1, \ldots, w_J)$ drawn from these corpora, the objective for an LLM parameterized by θ is the standard autoregressive log-likelihood,

$$\max_\theta \sum_{j=1}^{J} \log \pi_\theta(w_j \mid w_{<j}).$$

Optimizing this objective adapts the LLM to domain-specific terminology, concepts, and statistical associations. It does not directly supervise how the LLM reasons through a discovery problem. The post-training paradigms below act on the resulting LLM to improve the reasoning traces and actions it generates.

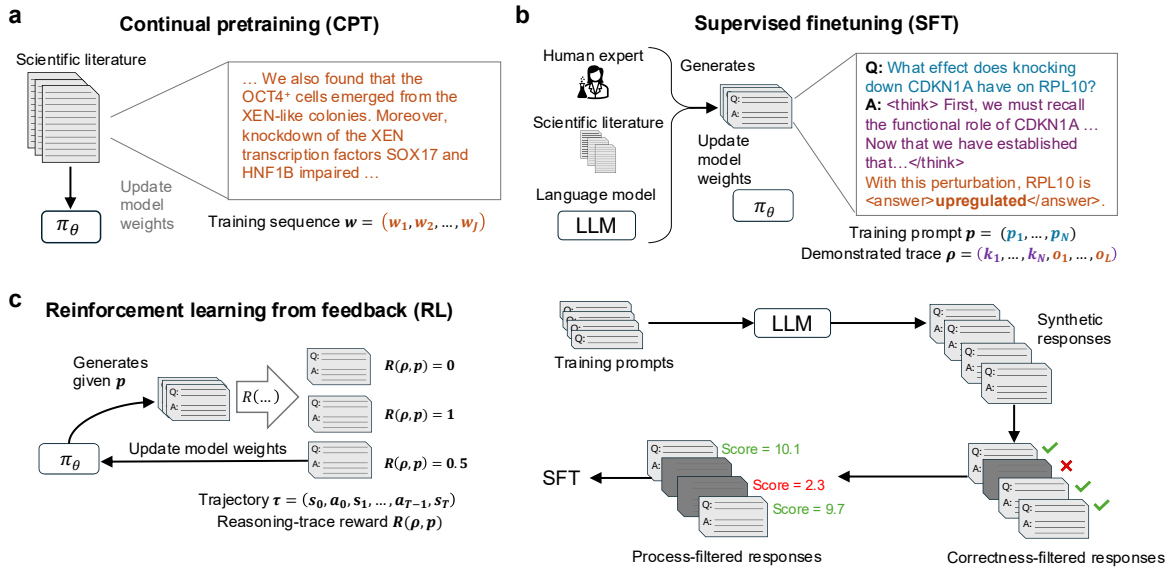

Figure 2. Training stages for scientific reasoning. (a) Continual pretraining (CPT) adapts the LLM to scientific knowledge by training it directly on text from relevant domain-specific literature using an autoregressive objective. **(b)** Supervised finetuning (SFT) trains the LLM to produce strong intermediate reasoning steps through behavioral cloning of expert-demonstrated, literature-derived, and synthetic reasoning traces (top). The success of SFT relies on high-quality training demonstrations, which can be obtained by filtering only demonstrations that result in a correct answer and/or contain a high-quality reasoning trace as determined by the scoring function (bottom). **(c)** Online reinforcement learning (RL) allows for the discovery of super-demonstration reasoning behavior under uncertainty via interactive exploration, using reward signals from evidence, interventions, and feedback.

Learning from Reasoning Traces

Reasoning *traces* (Box 1), sequences of generated intermediate steps that provide an explicit rationale for a hypothesis or selected action, provide a richer training signal than final answers alone. They supervise the generated rationale, including how the LLM decomposes a problem, integrates evidence across sources, and reasons under uncertainty. Traces come from expert demonstrations, reasoning steps extracted from scientific literature, and synthetic traces generated by LLMs themselves. These sources differ in quality, ease of collection, and the risks they introduce.

Supervised finetuning (SFT) [45,47] formalizes this supervision as imitation of high-quality demonstration traces (Figure 2b). Each entry in an SFT training dataset $\mathcal{D}$ consists of a prompt p

 doi:10.20944/preprints202608.2107.v1

and a target reasoning trace (Box 1) $\rho = (k_1, \ldots, k_N, o_1, \ldots, o_L)$, where k_n are intermediate reasoning tokens and o_l are final output tokens. For an LLM parameterized by θ, SFT maximizes the conditional likelihood of the demonstrated trace,

$$\max_{\theta} \mathbb{E}_{(p,\rho)\sim\mathcal{D}}[\log \pi_\theta(\rho \mid p)].$$

The LLM assigns high probability to each demonstrated reasoning and output token conditioned on the prompt and preceding tokens. In scientific settings, this encourages expert-like decomposition, evidence integration, and mechanistic interpretation, with the quality of the resulting behavior depending on the quality and coverage of the demonstrations.

Expert demonstrations drawn from established scientific workflows [2] can provide high-fidelity supervision, capturing hypothesis refinement and the role of prior mechanistic knowledge in interpreting ambiguous evidence. Curating them at the scale needed to train LLMs is prohibitive across most domains [44,48]. Literature-grounded traces scale better, since the methods, results, and discussion sections of a well-written paper provide a reconstructed account of how evidence was integrated and why particular conclusions were drawn, supplying supervision over mechanistic interpretation without additional annotation [4]. They inherit the bias of the published record described above, which overrepresents successful experiments and underrepresents the failed hypotheses and revised interpretations that constitute much of actual discovery.

Synthetic traces generated by distillation from stronger LLMs or by self-improvement [33,44,49] can scale further. When an LLM is trained on AI-generated rationales filtered by outcome correctness [44], it learns the reasoning patterns that correlate with correct answers in the training distribution, which need not correspond to valid inference, along with stylistic properties of the traces that contribute nothing to the inference. Such LLMs generate fluent but causally unsupported explanations, a failure mode with particular consequences in science, where a plausible and incorrect account directs experimental resources toward dead ends [50,51]. Overfitting to demonstration-specific artifacts can lead to catastrophic forgetting, where the LLM loses knowledge acquired during pretraining. Repeated training on LLM-generated data can separately produce model collapse, in which distributional diversity and performance on underrepresented inputs progressively degrade [52,53].

Process-based evaluation [48,54] filters traces based on the quality of the reasoning process as well as the correctness of the final answer, retaining only those whose intermediate steps satisfy metrics of mechanistic coherence and logical validity, thereby reducing the risk that the LLM learns to reach correct conclusions through spurious chains of reasoning (Figure 2b). Reasoning quality can be assessed against structured knowledge sources such as scientific knowledge graphs, or by an LLM that grades the trace against verbally described criteria (*e.g.*, "is the reasoning grounded in real biological mechanisms?") [55–57]. Both are soft verifiers applied to the reasoning process rather than to a hypothesis, and they carry the limitations we take up in the following section. Process-based filtering depends on a prior definition of a high-quality trace for the domain, which is often unavailable in biomedical discovery.

Reinforcement Learning from Feedback

Supervised learning imitates the reasoning behavior demonstrated in the training set through maximum likelihood, so it does not directly optimize the properties that make a trajectory scientifically useful unless those properties are encoded in the selection of demonstrations. Recent comparisons on verifiable tasks find that SFT stabilizes, expands, or replaces demonstrated behavior, whereas RL selectively amplifies capabilities accessible to the LLM and can sometimes improve out-of-distribution generalization [58–62].

Preprints.org (www.preprints.org) | NOT PEER-REVIEWED | Posted: 28 August 2026 doi:10.20944/preprints202608.2107.v1

Demonstration traces can be obtained for self-contained tasks, such as predicting the effect of a gene knockdown on a response gene. A discovery campaign, however, comprises a long sequence of dependent sub-tasks (Figure 3a), represented by the discovery trajectory defined in Box 1:

$$\tau = (s_0, a_0, s_1, a_1, \ldots, a_{T-1}, s_T).$$

At step t, the reasoning model generates a reasoning trace ρ_t and proposes an action a_t; executing that computational or experimental action returns the next observation s_{t+1}. For example, identifying an intervention for a disease requires the AI scientist to select a cell line, choose an initial panel of perturbations and a readout modality, and predict the effect of each perturbation on cell state. The AI scientist then ranks the candidates by their predicted effect, tests the ones it retains experimentally, and revises its beliefs in light of the resulting evidence. Although demonstration traces exist for many individual sub-tasks, few demonstrations capture the overarching discovery trajectory or the logic by which a verification budget is allocated across the entire campaign.

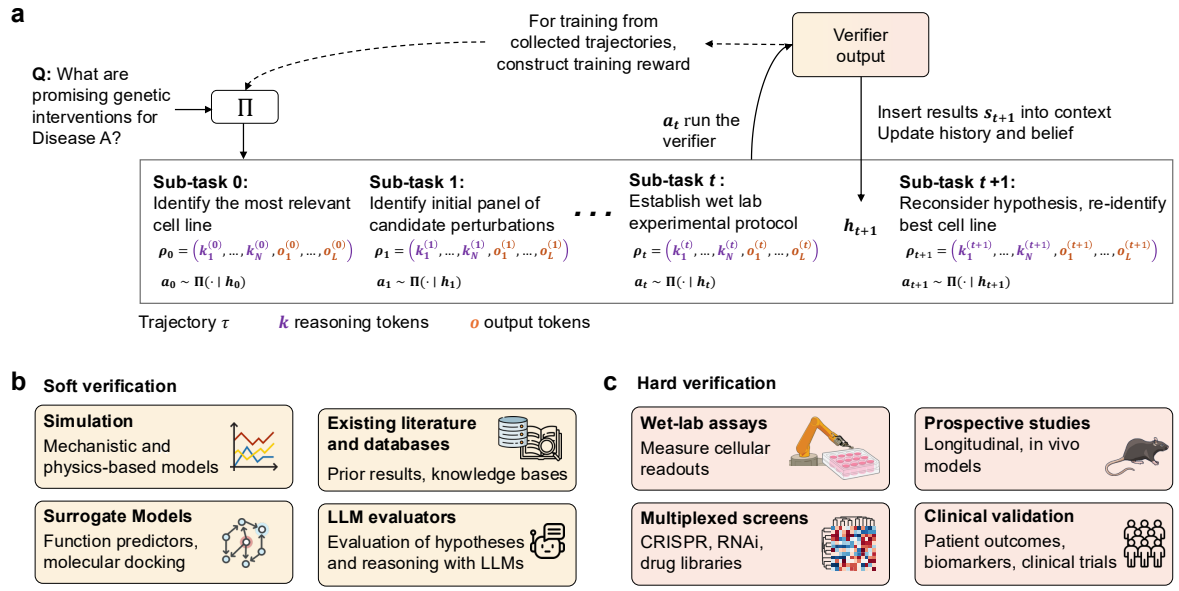

Figure 3. Verifying outputs of AI scientists. (a) Long-horizon scientific discovery trajectories comprise multiple dependent sub-tasks, making complete high-quality demonstrations difficult to obtain for SFT and credit assignment challenging when collected trajectories are used for RL. **(b)** Soft verifiers provide scalable computational verification through simulations, predictive surrogates, retrieval systems, and learned evaluators. **(c)** Hard verifiers return experimental observations through wet-lab assays, multiplexed screens, prospective studies, and clinical validation.

The campaign accounts that do exist, usually in scientific papers, are incomplete reconstructions rather than optimal discovery trajectories, so direct imitation alone provides no explicit mechanism for improving beyond them. At the level of an individual response, verifier-guided reinforcement learning addresses this limitation by optimizing model-generated reasoning traces against a specified reward (Figure 2c). At the campaign level, RL can instead optimize the AI scientist's action policy Π against the return accumulated over a discovery trajectory (Figure 3a). Neither form of RL guarantees scientific validity or performance beyond the available demonstrations. Whether RL creates new reasoning capacity or primarily elicits capabilities already latent in the base model remains debated [59,63].

Formally, in verifier-guided post-training, a prompt p is sampled from a prompt distribution $\mathcal{P}$, and the reasoning model samples a reasoning trace $\rho \sim \pi_\theta(\cdot \mid p)$. A scalar reward $R(\rho, p)$ is then constructed from verifier outputs and may encode properties such as correctness, mechanistic coherence, experimental actionability, expected information gain, or agreement with a verifier assessment. The objective is

$$\max_\theta \mathbb{E}_{p\sim\mathcal{P},\rho\sim\pi_\theta(\cdot|p)}[R(\rho, p)].$$

 doi:10.20944/preprints202608.2107.v1

This objective describes post-training over completed reasoning traces rather than the full experimental loop. Here, the constructed reward provides training feedback used to update the parameters θ. During a discovery campaign, by contrast, a verifier returns the next observation s_{t+1}; evidence derived from that output updates $(\mathcal{H}_t, b_t)$ to $(\mathcal{H}_{t+1}, b_{t+1})$, while θ remains fixed unless a separate parameter-learning procedure is invoked. Policy-gradient methods optimize the objective using baselines, advantage estimates, or group-relative normalization to reduce variance. Unlike SFT, which imitates a fixed demonstration distribution, RL can improve reasoning behavior when demonstrations are incomplete, suboptimal, or out of distribution, although this advantage has primarily been demonstrated on controlled, automatically verifiable tasks [58,60]. PPO- and GRPO-style on-policy methods are widely used in current reasoning post-training, where models improve across repeated attempts under constructed reward signals [64,65].

Learning Under Sparse and Delayed Feedback

Physical environments return observations at a cost and after a delay from the moment an AI output is generated, with noise arising from the limits of current experimental systems and from natural biological variation. Each observation therefore captures the state of a biological system only in part, and the rewards constructed from these observations can be sparse and difficult to assign to individual decisions. This structure separates scientific discovery from the domains where RL for reasoning has been most effective. In mathematics and programming, a rule-based verifier checks a completed candidate automatically and at low cost, so algorithms, such as GRPO [65], DAPO [66], and GSPO [67], can score many trajectories within a training run on verifiable tasks. Wet-lab feedback can be substantially slower and more costly than computational verification, and its latency, cost, and number of observations vary widely across assays and experimental designs. Reward design should account for the reduction in uncertainty produced by a negative or ambiguous observation. The multi-loop structure of a discovery campaign further complicates credit assignment, because a reward assigned at the end of the campaign can provide little information about which intermediate steps produced the outcome. Three algorithmic requirements follow from this feedback structure, and current RL methods satisfy them only in part.

(1) Long-horizon credit assignment [68], as the action that determines whether a campaign succeeds, such as which cell line to perturb, which readout to measure, or which confound to control, can precede the outcome on which its reward is eventually based by many loops, complicating attribution of the trajectory-level return to the earlier action. (2) Uncertainty-aware exploration [69], as allocating a limited verification budget across a campaign requires the AI scientist to maintain its belief b_t over the competing hypotheses $\mathcal{H}_t$ and to select the actions whose evidence can distinguish among them. (3) Adaptive planning under partial observability [70,71], as evidence updates $(\mathcal{H}_t, b_t)$ and changes which actions are worth taking in later loops, so the AI scientist must revise its plan rather than execute a plan formed under a belief the evidence has since superseded.

Current RL methods address these requirements at different levels. Hierarchical RL introduces temporal abstractions or sub-task policies that separate local reasoning within a sub-task from the synthesis of sub-task conclusions, which can help localize credit across a long reasoning trace or discovery trajectory and therefore bears on Requirement (1) [72–74]. Bayes-adaptive RL addresses Requirement (2) by maintaining the uncertainty represented in the belief and selecting actions according to their value for both information gathering and achieving the scientific objective [69].

At the campaign level, model-based RL uses a world model p_ϕ to predict the latent state z_{t+1} and the resulting observation s_{t+1} under candidate actions a_t, and plans using those predictions, which reduces how often a campaign interacts with the physical environment [27,75–77]. Perturbation response predictors provide components of such a world model, for example, by predicting the effect of a perturbation on gene expression [78,79]. When evidence derived from the observed s_{t+1} is inconsistent with the current hypothesis set and belief $(\mathcal{H}_t, b_t)$, the AI scientist can update to $(\mathcal{H}_{t+1}, b_{t+1})$ and replan, which satisfies Requirement (3). Updating or recalibrating the parameters of the world model is a separate learning problem performed on observations accumulated within

 doi:10.20944/preprints202608.2107.v1

or across campaigns. However, planning can be vulnerable to errors in the world model, and an improvement in predictive performance does not necessarily produce a better plan. Reward design can shape these behaviors during training, but meeting the three requirements also depends on how the AI scientist represents beliefs and uncertainty, explores among competing hypotheses, and plans across loops.

Reward Design over Reasoning Traces and Experiments

Standard benchmark optimization maximizes accuracy on held-out questions with reference answers, which is a poor proxy for the properties required in a discovery campaign [80–82], since many consequential scientific problems admit no unique or immediately verifiable solution. A reasoning model should therefore be assessed at the level of both its output and its reasoning trace. The criteria are whether a generated hypothesis is mechanistically coherent, experimentally actionable, causally consistent with known biology, and robust to distributional shifts across experimental systems, and whether the accompanying trace integrates relevant evidence and uncertainty [83–86]. These criteria concern the reasoning model's output for an individual task. Whether the complete AI scientist then selects an informative experiment, interprets the resulting observation appropriately, and revises its beliefs in light of the evidence is a campaign-level property evaluated separately below.

These requirements motivate reward signals that go beyond answer correctness. Process-level rewards [48,87] score intermediate steps in a reasoning trace rather than only its final output, which is useful in scientific domains where answer correctness alone does not establish that the generated rationale is mechanistically coherent. Assessments from computational models of biological systems [10] can provide an additional signal by testing whether a hypothesis satisfies quantitative constraints imposed by known pathway dynamics, stoichiometry, or evolutionary conservation. A third class of reward evaluates the experiment implied by a hypothesis according to its feasibility, expected ability to discriminate among competing accounts, and expected reduction in uncertainty [6]. These rewards are scalar training signals constructed from verifier assessments or predicted campaign utility. They are not the observations returned by experiments themselves.

Verification of AI Outputs Across Discovery Horizons

The outputs on which a discovery campaign acts include hypotheses, proposed experiments and assay plans, the protocols that implement those plans, interpretations of the resulting observations, and belief updates made in light of the evidence. Verifiers return computational results or experimental measurements. After quality control and appropriate statistical or mechanistic analysis, these outputs provide evidence for the AI output under evaluation. Evidence is therefore distinct both from the raw verifier output and from the subsequent interpretation placed on it. For claim-specific evidence to count as prospective verification, the claim and evaluation criterion must be specified before the verifier is queried. Generating and then assessing a hypothesis with the same model does not constitute independent verification. Soft and hard verifiers (Box 1) are distinguished by the source of their evidence, computational evaluation versus physical experimental measurement, rather than by a fixed boundary in cost or reliability. Computational results often arrive sooner and cover more candidates, whereas physical measurements interrogate the system more directly. Either can be noisy, biased, or uninformative for the claim at hand.

Soft Verification of AI Outputs as Approximate Guidance at Scale

Soft verifiers return computational results of AI outputs. Some act as predictive surrogates for physical experiments; these include learned structure and function predictors, mechanistic simulations [88,89], and molecular docking models [19,90,91] (Figure 3b). Others, including knowledge retrieval systems [92], assess consistency with existing evidence rather than approximating an experimental outcome (Figure 3b). These verifiers need not return a single deterministic score. They may produce probability distributions, structured predictions, simulated trajectories, or retrieved evidence spanning multiple data modalities. When their outputs are thresholded to advance candidates, the

 doi:10.20944/preprints202608.2107.v1

resulting false-discovery rate depends on the operating point and candidate distribution; calibration and domain validity can likewise vary across the hypothesis space, particularly in regions underrepresented in training data [93–96]. A verifier output should therefore be interpreted within the domain and decision threshold for which it was validated, together with its uncertainty [97,98].

Soft verifiers vary substantially in computational cost, and current workflows often evaluate candidates in stages, progressing from cheaper, high-throughput filters to more expensive, higher-fidelity analyses. In materials discovery, for example, GNoME uses learned energy models to filter candidate crystal structures before evaluating selected candidates using density functional theory [99]. Protein engineering pipelines similarly combine multiple verifiers, including AlphaFold-family confidence and interface metrics, Rosetta-derived binding energies, and geometric proxies such as buried surface area and the number of interfacial hydrogen bonds, to corroborate evidence across distinct computational objectives [100–102]. More computationally intensive analyses, such as molecular dynamics, can then be applied to a smaller subset of candidates to assess their stability and conformational dynamics [89]. Combining multiple verifiers, however, introduces a multi-objective optimization problem, as their scores may conflict or trade off against one another [103]. Nor does every domain yet have a sufficiently reliable predictive soft verifier; for example, accurately forecasting transcriptional responses to genetic or chemical perturbations in unseen cellular contexts and disease states remains challenging [95,96].

The choice and combination of soft verifiers should be refined as experimental evidence accumulates. In protein-complex prediction, for example, interface-focused metrics such as actifpTM [104] and ipSAE [105] were developed to correct biases in ipTM. When heterogeneous verifiers measure distinct objectives, their trade-offs should be retained explicitly, for example as a Pareto set, rather than concealed in an arbitrary weighted sum [103]. When they estimate the same outcome, their scores should be calibrated against held-out experiments, and agreement among verifiers that share training data or modeling assumptions should not be treated as independent corroboration. Disagreement, missing coverage, or high predictive uncertainty should trigger an experiment selected to distinguish among the alternatives. Representative experimental outcomes can then update each verifier's weight within the domain in which it has demonstrated predictive validity [97,98,106].

Finally, sufficiently reliable soft verifiers can turn some scientific tasks into scalable sources of synthetic supervision for post-training reasoning models. Candidate reasoning traces can be sampled and filtered using verifier scores to construct supervised finetuning data, while verifier outputs can be used to construct outcome- or process-level rewards for reinforcement learning, linking soft verification to the SFT and RL approaches described above [14,107,108]. In biology, RBIO1 provides a proof of concept by using predictions from world models and structured prior knowledge as rewards for reinforcement learning [109]. Optimization against a soft verifier can improve its scores without improving performance on experimental outcomes [110].

Hard Verification of AI Outputs Through Experimental Grounding

Hard verifiers provide empirical observations from physical systems, including biochemical binding and activity assays [102,111,112], pooled cellular perturbation screens [29,30], automated synthesis and characterization [7,113], and prospective validation in organoid and clinical systems [10] (Figure 3c). Unlike soft verifiers, they interrogate the biological system and can reveal constraints that computational surrogates do not capture. However, experiments test observable predictions derived from a hypothesis conditional on assumptions about the model system, assay, controls, and measurement process. Their interpretation therefore depends on assay validity, appropriate controls, statistical power, measurement noise, and reproducibility across experimental contexts [114,115]. Experimental results should consequently be treated as more direct, but still conditional and potentially ambiguous, evidence. A null result is informative on the same terms, since it can reduce uncertainty over competing mechanisms without establishing any of them, and it constrains a hypothesis only as far as the assay was powered to detect the predicted effect.

Whereas a theorem prover can evaluate large numbers of candidate proofs computationally [13], an AI scientist can test only a small fraction of its hypotheses experimentally. Experiments should

 doi:10.20944/preprints202608.2107.v1

therefore be prioritized by their expected scientific utility, namely whether they distinguish competing mechanisms, reduce uncertainty, or expose failures in the current world model, rather than only by their likelihood of supporting a candidate. Active learning methods select informative interventions under a constrained verification budget [116,117]. Batched designs relax the constraint, since multiplexed screens and pooled perturbation assays test many hypotheses in one execution, and the problem becomes one of choosing sets of experiments that are jointly informative rather than ranking them individually [118].

De novo protein-binder validation illustrates why hard verification requires multiple forms of evidence. Biolayer interferometry (BLI) and surface plasmon resonance (SPR) estimate binding affinity and kinetics by fitting measured association and dissociation curves [102,111,112]. However, an apparently convincing sensorgram can result from nonspecific surface adsorption, avidity, aggregation, or other assay artifacts rather than a specific one-to-one interaction [119]. Nor does binding alone establish that a design engages the intended epitope or produces the proposed biological mechanism. Confidence can instead accumulate across experiments when they address distinct alternative explanations, combining appropriate controls with competition or epitope-binning experiments, mutagenesis of predicted interface residues, orthogonal functional assays, and structural determination by cryo-electron microscopy or X-ray crystallography [120]. Hard verifiers therefore do not return definitive binary labels; they provide graded and complementary observations whose evidential meaning depends on assay validity and interpretation. A measurement can be technically reliable while remaining consistent with several biological explanations.

Multi-Loop Discovery Campaigns

Closed-loop discovery requires the AI scientist to combine evidence derived from computational results and experimental observations over many loops. At each loop, the AI scientist maintains a belief b_t over the current competing hypotheses $\mathcal{H}_t$. It then selects an action a_t expected to reduce uncertainty in b_t or discriminate among hypotheses in $\mathcal{H}_t$, or reveal evidence that motivates new or revised hypotheses, while respecting the constraints of the campaign. Executing a_t through a computational or experimental verifier produces the next observation s_{t+1}. Evidence derived from s_{t+1} updates $(\mathcal{H}_t, b_t)$ to $(\mathcal{H}_{t+1}, b_{t+1})$ and thereby changes which actions are valuable in subsequent loops.

When the represented hypothesis set and probabilistic model are held fixed, Bayesian experimental design formalizes action selection as a sequential decision problem in which each action is evaluated according to its expected utility under the current belief b_t [34,35,121]. Because the outcome of an experiment is not known in advance, this utility is averaged over the possible outcomes predicted by the current model. When the goal is to discriminate among competing hypotheses, a natural utility is expected information gain, which measures the expected reduction in uncertainty in b_t [35,121]. These formulations provide a principled framework for experiment selection, but they do not completely solve multi-loop discovery. They optimize only over the hypotheses, actions, models, and utilities represented in the system, so an information-gain objective may efficiently discriminate among hypotheses in the current set $\mathcal{H}_t$ while assigning no value to explanations outside that set. This limitation is particularly important in open-ended discovery, where an observation can be valuable because it motivates an expansion or revision of $\mathcal{H}_t$, rather than because it merely reduces uncertainty in b_t. The resulting policy is also only as reliable as its probabilistic model and uncertainty estimates, since model misspecification can produce misleading uncertainty and direct exploration toward suboptimal experiments [122]. Because actions can differ in how long they take to complete, and because the value of one experiment can depend on the experiments that follow it, practical campaigns may require asynchronous batch or non-myopic policies that account for pending measurements and the future value of the remaining budget [35,123].

Self-driving laboratories instantiate a constrained form of this closed-loop framework. They generally hold the experimental protocol and readout fixed while varying a pre-defined set of experimental parameters, such as the temperature or reagent concentrations used to synthesize a target molecule [7,113,124]. An instrument then measures a predefined outcome and automated analysis

 doi:10.20944/preprints202608.2107.v1

determines how well that outcome satisfies the objective of the campaign. Because the experimental protocol, action space, and success criteria are largely specified in advance, many loops can proceed under human-on-the-loop oversight. The A-Lab follows this pattern by combining computational screening, literature-derived synthesis knowledge, active learning, robotic experimentation, and automated characterization to search for synthesis routes to inorganic materials over repeated loops [7]. Outcomes from unsuccessful syntheses influenced subsequent synthesis choices in the same campaign, while retrospective analysis identified computational and experimental failure modes that could inform future systems. Related closed-loop strategies have been used for Bayesian optimization in chemistry and materials discovery, as well as for protein engineering, where sequence models are updated using fitness measurements collected over successive design, build, test, and learn loops [113,125–128].

Allocating hypotheses to verifiers becomes a difficult selection problem when an AI scientist can select among many verifiers with different costs, fidelities, and completion times. In this setting, an action a_t encodes both which candidate is tested and which verifier is queried. Experiment selection must therefore weigh the expected value of the information returned by an action against its cost, completion time, material requirements, computational demand, instrument capacity, reagent availability, and other relevant resource constraints. Multi-fidelity Bayesian optimization formalizes this problem by modeling relationships among information sources that differ in predictive fidelity and cost, then using an acquisition function to choose which candidate and which information source to query [106,129]. In AI scientists, such sources can include simulations, soft verifiers, proxy assays, and wet-lab experiments when their relationships to the hypothesis under consideration in the current loop can be modeled. A multi-fidelity model that is jointly fitted over observations from all sources transfers information among them, and the acquisition function can trade off expected information gain against the cost of each observation [106,130]. A soft verifier is useful when it predicts the outcome of the experiment that tests the hypothesis under consideration. When it is poorly correlated with that outcome, repeatedly querying it misdirects the campaign and costs more than directly testing the hypothesis [106,131].

Evaluation of Closed-Loop Discovery

Comprehensive evaluation of closed-loop AI scientists assesses both the intermediate steps within each loop and the discovery trajectory τ across a campaign. It requires criteria that measure whether the AI scientist selects experiments that distinguish among competing hypotheses, revises those hypotheses as warranted by the evidence returned in each loop, and reaches a correct final answer. These criteria evaluate the AI scientist as a whole because campaign performance depends jointly on the reasoning model, the harness that orchestrates its actions, and the verifiers that supply feedback. An evaluation should therefore specify the version and configuration of each component at the start of the campaign and report whether any component was allowed to change during it.

We consider three complementary evaluation settings. Retrospective and prospective evaluation are distinguished by when the evaluation evidence becomes available relative to the run. Retrospective evaluation compares the AI scientist's output against a result recorded before the run began, so the evaluation evidence is fixed and cannot be influenced by the system's actions. Prospective evaluation compares the output against a measurement collected only after the AI scientist has committed to a prediction. Loop-level evaluation instead assesses adaptation across repeated rounds of feedback, evaluating how evidence received in one loop changes the hypotheses and actions selected in subsequent loops.

Retrospective benchmarks and rediscovery evaluation. Retrospective evaluation constructs tasks from historical experimental data by treating known results as held-out targets (Figure 4a). In a rediscovery setting, an AI scientist recovers a published finding, such as a drug-target interaction or a gene-phenotype association, from the evidence available before its original report. This tests hypothesis generation under a controlled information constraint [4,10]. Retrospective benchmarks can also score the procedure that produces an answer rather than the answer itself, including the predictive

 doi:10.20944/preprints202608.2107.v1

models an AI scientist queries and the computational analyses it performs. For example, BixBench defines tasks from published bioinformatics analyses in which an agent retrieves the data, writes and runs code, and returns an answer scored against the recorded result [132]. CompBioBench modifies real datasets and removes their metadata to define tasks with a single answer that cannot be reached by recalling a published conclusion, and frontier agents answer most of them correctly [133]. Agentic benchmarks evaluate the execution of an analysis rather than its conclusion, for example by varying the number of dependent steps a task requires, and agents complete a small fraction of the longest tasks that graders score [134]. These benchmarks show that current AI scientists perform well on tightly scoped analyses and poorly on open-ended tasks composed of many dependent steps, which is the structure of a closed-loop discovery campaign. Tasks of this kind return open-ended outputs, and scoring them requires rubrics that capture answer accuracy, task completeness, and validity of the procedure that produced the answer [135]. These benchmarks evaluate multi-step tasks retrospectively rather than multi-loop campaigns, since the recorded result is fixed before the run and no verifier returns evidence during it that could revise the hypotheses the AI scientist holds or the action it selects next.

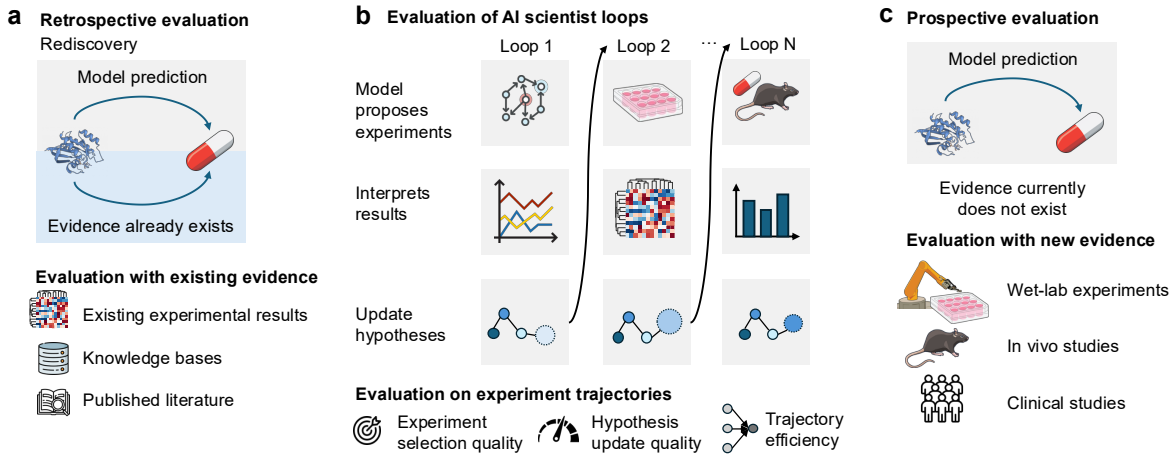

Figure 4. Evaluation settings for AI scientists. (a) Retrospective evaluation uses historical datasets, held-out discoveries, perturbation tasks, and literature-derived benchmarks. **(b)** Evaluation of AI scientist loops measures adaptation across repeated loops of feedback and experiment selection. **(c)** Prospective evaluation tests scientific reasoning through wet-lab experiments, robotic platforms, clinical studies, and real-world deployment.

Retrospective benchmarks require an answer based on previously recorded evidence, so such a benchmark can include only tasks for which a past research study reports the target result. The targets can also appear in the corpus the LLM was pretrained on, so a performance score can reflect retrieval of a memorized result rather than inference from the evidence the task provides, unless tasks are separated by publication date or drawn from findings that postdate the training cutoff [136]. While retrospective evaluation tests whether an AI scientist reasons correctly over historical data, it cannot establish that the AI scientist selects an experiment that resolves an open question. Selecting such an experiment requires a prediction about the outcome of a loop that no campaign has yet run, so no historical record can score it.

Evaluation of AI scientist loops. Evaluation of loops assesses reasoning over the trajectory of multiple loops in which an AI scientist receives feedback and adapts its strategy (Figure 4b). Within a loop, the AI scientist proposes an experiment, receives an output from a computational or experimental verifier, interprets the evidence derived from that output, and updates its hypotheses before selecting the next experiment. This sequence tests the multi-loop reasoning that retrospective benchmarks cannot probe [137]. Because the object under evaluation is the discovery trajectory τ rather than a single answer, performance metrics in this setting consider final answer accuracy as well as the quality of the intermediate steps, whether the AI scientist selects the experiment that separates competing hypotheses in each loop, whether its uncertainty estimates are calibrated to the evidence it has received, and

 doi:10.20944/preprints202608.2107.v1

whether it recovers from results that contradict its current hypothesis. The conclusions this evaluation setting supports depend on where the feedback originates. A reasoning model trained against a simulator can learn to optimize the simulator rather than reason about the biological system, and evaluating it against that same simulator will not detect the substitution [138]. Feedback derived from datasets that overlap its pretraining corpus likewise makes adaptation appear stronger than it is [139,140]. A learned or approximate verifier used to train an AI scientist should therefore not also serve as its sole evaluator. Evaluation should rely on leakage-free targets that provide an independent test of performance.

Prospective evaluation. Prospective evaluation tests predictions against experiments that have not yet been run when the prediction is made, which rules out an AI scientist retrieving or recalling the outcome from its training dataset (Figure 4c) [6,10,113]. Establishing novelty additionally requires determining whether the finding is already supported by the evidence available to the AI scientist. Current prospective evaluations typically report a handful of AI outputs and are rarely preregistered. Preregistration [141] addresses this selection bias by fixing which AI outputs will be verified before experiments are run. A campaign can register the hypotheses to be tested, the experiments selected to test them, and the criteria under which a readout counts as successful verification, then report the fraction of tested hypotheses that held up and the cost of those tests. The registration record is necessary because a campaign that reports only the hypotheses it selected after seeing which ones succeeded will overstate success rate. Clinical research has widely adopted preregistration [142], and closed-loop discovery needs it as well. Prospective evaluation can show which AI outputs held up, not the rate at which AI scientists generate sound outputs, so retrospective benchmarks and evaluations of loops are necessary for the breadth of testing that biological and clinical laboratories cannot perform.

Outlook

Current campaigns of AI scientists are short in duration and few in the number of loops, and the hypotheses that reach experimental testing are selected by scientists. Scientists retain substantial oversight over which hypotheses and experiments are pursued, and only a small fraction of generated outputs reach prospective validation or yield novel scientific insights. Moving past such demonstrations will require methods that connect a generated hypothesis to the measurement designed to test it, derive evidence from that measurement, and transfer that evidence to the next stage of discovery. Developing these methods will require adapting AI scientists to feedback that varies in timescale and cost. Many RL-trained reasoning models are optimized against dense and immediately verifiable rewards [14,143], whereas experimental results in a closed-loop campaign arrive over longer and more variable timescales, and AI scientists must separate predicted effects from biological and technical variations. Soft verifiers whose agreement with experimental outcomes has been established in early loops reduce how often a campaign escalates to hard verification [106]. Miniaturized, multiplexed, and automated assays can shorten the interval between perturbation and an interpretable experimental readout, thereby reducing the latency of hard verification [144]. However, hard verifiers introduce problems of reproducibility and biosafety (Box 2) with no counterpart among computational soft verifiers [6,113].

Retrospective benchmarks are essential for developing closed-loop AI scientists, but repeated use makes them vulnerable to contamination, overfitting, and saturation, which limits what they establish about scientific discovery. Repeated evaluation can lead to overfitting of reasoning models, prompts, and harnesses to a fixed benchmark [145,146]. Their useful lifetime is therefore short, since scores on recently introduced benchmarks can rise by tens of percentage points within a year and tasks approach saturation [147,148]. These limitations do not remove the need for retrospective evaluation, but instead motivate continually renewed benchmarks built from temporally held-out, contamination-resistant, or dynamically generated tasks. Evaluation of the closed loop should complement these benchmarks by testing whether an AI scientist revises hypotheses, selects informative next experiments, and recovers from failed predictions as new evidence arrives, although simulated or replayed feedback cannot

 doi:10.20944/preprints202608.2107.v1

establish that these behaviors transfer to a real-world biological system. As closed-loop AI systems mature, evaluation may resemble the research process, with loops of retrospective evaluations followed by prospective tests on new questions.

The promise of closed-loop biomedical discovery also raises many questions. The history accumulated across loops has to be maintained over months, at a cost that grows with the length of the campaign, and the training and inference behind AI scientists require computing infrastructure alongside material and instrument costs for running discovery loops. Both computational and experimental safeguards must be implemented, as an incorrect conclusion acted on by a self-driving laboratory can direct months of laboratory work before the error becomes apparent and can also lead to biosecurity risks [38]. Open datasets and shared platforms for AI scientists are a further consideration, since engineering expertise is concentrated in a small number of institutions [4]. Autonomous discovery campaigns that laboratories can join at any point, contributing measurements to a continuing effort, would change how collaborations form across science. Scientific knowledge would then accumulate through a continuous exchange between hypothesis and measurement in which human scientists and AI scientists both take part.

Box 2 | Societal considerations for closed-loop AI-driven biomedical discovery

Democratization of AI scientist capability. Advanced experimental capability exists at well-resourced institutions, and AI scientists could widen access to it by giving researchers worldwide the means to run discovery loops [4,149]. These systems rely on resources that exist at a small number of institutions, since training a reasoning model requires computing infrastructure, proprietary experimental datasets, and engineering expertise, while hard verification requires instruments, reagents, and laboratory staff. Open datasets, shared computing infrastructure, and shared experimental facilities would make closed-loop AI scientists available more widely.

Scientific hallucination, auditing of discovery loops, and reproducibility. A loop that acts on a hypothesis that is plausible but mechanistically unsupported can commit a laboratory to months of unproductive work. Recognizing the failure requires an audit of the reasoning trajectory, so a campaign has to keep a record of each hypothesis, the action chosen to test it, and the evidence the verifier returned, together with the model version and the harness configuration. A stochastic policy returns a different trajectory on a second run of the same question, and this is compounded by the natural biological variation of assays [114], so reproducing a closed-loop campaign is harder than reproducing the policy or the assay on its own.

Biosecurity and safety risks. Closed-loop AI scientists are dual-use systems, since a system with tool access and a route to synthesis can be directed toward molecules of concern as readily as toward a therapeutic target. A campaign also divides an objective across many loops, so each individual request can appear innocuous [38]. Useful controls include restricting the action space, requiring approval for actions that raise biosafety concerns, screening synthesis orders, and keeping trajectories for later review, and they can be implemented in the AI scientist harness rather than in the reasoning model [43].

Evolving roles of human researchers and AI scientists. As AI scientists take on hypothesis generation, experimental planning, and data interpretation, scientific expertise will center on mechanistic judgment, experimental oversight, and evaluation of AI outputs. Research institutions will have to reconsider how they train students, institutions how they hire, and the community how it assigns academic credit [150].

Author Contributions: All authors contributed to the design and writing of the manuscript, helped shape the research, provided critical feedback, and commented on the manuscript and its revisions. M.Z. conceived the study and was in charge of overall direction and planning.

 doi:10.20944/preprints202608.2107.v1

Acknowledgments: We gratefully acknowledge support, in part, by NSF CAREER 2339524, ARPA-H Biomedical Data Fabric (BDF) Toolbox Program, Amazon Faculty Research, Google Research Scholar Program, AstraZeneca Research, Roche Alliance with Distinguished Scientists (ROADS) Program, Sanofi iDEA-iTECH Award, GlaxoSmithKline Award, Boehringer Ingelheim Award, Merck Award, Optum AI Research Collaboration Award, Pfizer Research, Gates Foundation (INV-079038), Chan Zuckerberg Initiative, John and Virginia Kaneb Fellowship at Harvard Medical School, Biswas Computational Biology Initiative in partnership with the Milken Institute, Harvard Medical School Dean's Innovation Fund for the Use of Artificial Intelligence, and the Kempner Institute for the Study of Natural and Artificial Intelligence at Harvard University. This work was delivered as part of the AURORA project supported by the Cancer Grand Challenges partnership funded by Cancer Research UK ([CGCAI1-Mar26/100001]). A.N. is supported by the Rhodes Scholarship. A.F. and L.F. are supported by the Kempner Graduate Fellowship at Harvard University. Any opinions, findings, conclusions, or recommendations expressed in this material are those of the authors and do not necessarily reflect the views of the funders.

Conflicts of Interest: The authors declare no competing interests.

References

1. Wang, H.; Fu, T.; Du, Y.; Gao, W.; Huang, K.; Liu, Z.; Chandak, P.; Liu, S.; Van Katwyk, P.; Deac, A.; et al. Scientific discovery in the age of artificial intelligence. *Nature* **2023**, *620*, 47–60.
2. Gao, S.; Fang, A.; Huang, Y.; Giunchiglia, V.; Noori, A.; Schwarz, J.R.; Ektefaie, Y.; Kondic, J.; Zitnik, M. Empowering biomedical discovery with AI agents. *Cell* **2024**, *187*, 6125–6151.
3. M. Bran, A.; Cox, S.; Schilter, O.; Baldassari, C.; White, A.D.; Schwaller, P. Augmenting large language models with chemistry tools. *Nature Machine Intelligence* **2024**, *6*, 525–535.
4. Ifargan, T.; Hafner, L.; Kern, M.; Alcalay, O.; Kishony, R. Autonomous LLM-driven research—from data to human-verifiable research papers. *NEJM AI* **2025**, *2*, AIoa2400555.
5. Mitchener, L.; Yiu, A.; Chang, B.; Bourdenx, M.; Nadolski, T.; Sulovari, A.; Landsness, E.C.; Barabasi, D.L.; Narayanan, S.; Evans, N.; et al. Kosmos: An AI scientist for autonomous discovery. *arXiv:2511.02824* **2025**.
6. Boiko, D.A.; MacKnight, R.; Kline, B.; Gomes, G. Autonomous chemical research with large language models. *Nature* **2023**, *624*, 570–578.
7. Szymanski, N.J.; Rendy, B.; Fei, Y.; Kumar, R.E.; He, T.; Milsted, D.; McDermott, M.J.; Gallant, M.; Cubuk, E.D.; Merchant, A.; et al. An autonomous laboratory for the accelerated synthesis of inorganic materials. *Nature* **2023**, *624*, 86–91. https://doi.org/10.1038/s41586-023-06734-w.
8. Gottweis, J.; Weng, W.H.; Daryin, A.; et al. Accelerating Scientific Discovery with Co-Scientist. *Nature* **2026**, *655*, 487–496. https://doi.org/10.1038/s41586-026-10644-y.
9. Ghareeb, A.E.; Chang, B.; Mitchener, L.; Yiu, A.; Szostkiewicz, C.J.; Shved, D.; Gyimesi, G.J.; Laurent, J.M.; Wright, S.M.; Razzak, M.T.; et al. A multi-agent system for automating scientific discovery. *Nature* **2026**, *655*, 497–505. https://doi.org/10.1038/s41586-026-10652-y.
10. Noori, A.; Polonuer, J.; Meyer, K.; Budnik, B.; Morton, S.; Wang, X.; Nazeen, S.; He, Y.; Arango, I.; Vittor, L.; et al. Graph AI generates neurological hypotheses validated in molecular, organoid, and clinical systems. *arXiv:2512.13724* **2025**.
11. Hubert, T.; Mehta, R.; Sartran, L.; Horváth, M.Z.; Žužić, G.; Wieser, E.; Huang, A.; Schrittwieser, J.; Schroecker, Y.; Masoom, H.; et al. Olympiad-level formal mathematical reasoning with reinforcement learning. *Nature* **2026**, *651*, 607–613. https://doi.org/10.1038/s41586-025-09833-y.
12. Ren, Z.; Shao, Z.; Song, J.; Xin, H.; Wang, H.; Zhao, W.; Zhang, L.; Fu, Z.; Zhu, Q.; Yang, D.; et al. Deepseek-prover-v2: Advancing formal mathematical reasoning via reinforcement learning for subgoal decomposition. *arXiv:2504.21801* **2025**.
13. Kumarappan, A.; Tiwari, M.; Song, P.; George, R.J.; Xiao, C.; et al. Leanagent: Lifelong learning for formal theorem proving. In Proceedings of the International Conference on Learning Representations, 2025, Vol. 2025, pp. 73525–73564.
14. Guo, D.; Yang, D.; Zhang, H.; Song, J.; Wang, P.; Zhu, Q.; Xu, R.; Zhang, R.; Ma, S.; Bi, X.; et al. DeepSeek-R1 incentivizes reasoning in LLMs through reinforcement learning. *Nature* **2025**, *645*, 633–638. https://doi.org/10.1038/s41586-025-09422-z.
15. Team, K.; Du, A.; Gao, B.; Xing, B.; Jiang, C.; Chen, C.; Li, C.; Xiao, C.; Du, C.; Liao, C.; et al. Kimi k1. 5: Scaling reinforcement learning with LLMs. *arXiv:2501.12599* **2025**.

 doi:10.20944/preprints202608.2107.v1

16. Romera-Paredes, B.; Barekatain, M.; Novikov, A.; Balog, M.; Kumar, M.P.; Dupont, E.; Ruiz, F.J.; Ellenberg, J.S.; Wang, P.; Fawzi, O.; et al. Mathematical discoveries from program search with large language models. *Nature* **2024**, *625*, 468–475.
17. Novikov, A.; Vũ, N.; Eisenberger, M.; Dupont, E.; Huang, P.S.; Wagner, A.Z.; Shirobokov, S.; Kozlovskii, B.; Ruiz, F.J.; Mehrabian, A.; et al. AlphaEvolve: A coding agent for scientific and algorithmic discovery. *arXiv preprint arXiv:2506.13131* **2025**.
18. Abramson, J.; Adler, J.; Dunger, J.; Evans, R.; Green, T.; Pritzel, A.; Ronneberger, O.; Willmore, L.; Ballard, A.J.; Bambrick, J.; et al. Accurate structure prediction of biomolecular interactions with AlphaFold 3. *Nature* **2024**, *630*, 493–500.
19. Avsec, Ž.; Latysheva, N.; Cheng, J.; Novati, G.; Taylor, K.R.; Ward, T.; Bycroft, C.; Nicolaisen, L.; Arvaniti, E.; Pan, J.; et al. Advancing regulatory variant effect prediction with AlphaGenome. *Nature* **2026**, *649*, 1206–1218.
20. Brixi, G.; Durrant, M.G.; Ku, J.; Naghipourfar, M.; Poli, M.; Sun, G.; Brockman, G.; Chang, D.; Fanton, A.; Gonzalez, G.A.; et al. Genome modelling and design across all domains of life with Evo 2. *Nature* **2026**, *652*, 1349–1361.
21. Pearce, J.D.; Simmonds, S.E.; Mahmoudabadi, G.; Krishnan, L.; Palla, G.; Istrate, A.M.; Tarashansky, A.; Nelson, B.; Valenzuela, O.; Li, D.; et al. TranscriptFormer: A generative cell atlas across 1.5 billion years of evolution. *Science* **2026**, *393*, aec8514, [https://www.science.org/doi/pdf/10.1126/science.aec8514]. https://doi.org/10.1126/science.aec8514.
22. Hao, Q.; Xu, F.; Li, Y.; Evans, J. Artificial intelligence tools expand scientists' impact but contract science's focus. *Nature* **2026**, *649*, 1237–1243.
23. Hu, Y.; Liu, S.; Yue, Y.; Zhang, G.; Liu, B.; Zhu, F.; Lin, J.; Guo, H.; Dou, S.; Xi, Z.; et al. Memory in the Age of AI Agents, 2026, [arXiv:cs.CL/2512.13564].
24. Fanelli, D. Negative results are disappearing from most disciplines and countries. *Scientometrics* **2012**, *90*, 891–904.
25. Fanelli, D. "Positive" results increase down the hierarchy of the sciences. *PloS one* **2010**, *5*, e10068.
26. Turner, E.H.; Matthews, A.M.; Linardatos, E.; Tell, R.A.; Rosenthal, R. Selective publication of antidepressant trials and its influence on apparent efficacy. *New England Journal of Medicine* **2008**, *358*, 252–260.
27. Ha, D.; Schmidhuber, J. Recurrent World Models Facilitate Policy Evolution. In Proceedings of the Advances in Neural Information Processing Systems; Bengio, S.; Wallach, H.; Larochelle, H.; Grauman, K.; Cesa-Bianchi, N.; Garnett, R., Eds. Curran Associates, Inc., 2018, Vol. 31.
28. Hafner, D.; Pasukonis, J.; Ba, J.; Lillicrap, T. Mastering diverse control tasks through world models. *Nature* **2025**, *640*, 647–653. https://doi.org/10.1038/s41586-025-08744-2.
29. Dixit, A.; Parnas, O.; Li, B.; Chen, J.; Fulco, C.P.; Jerby-Arnon, L.; Marjanovic, N.D.; Dionne, D.; Burks, T.; Raychowdhury, R.; et al. Perturb-seq: Dissecting molecular circuits with scalable single-cell RNA profiling of pooled genetic screens. *Cell* **2016**, *167*, 1853–1866.e17.
30. Replogle, J.M.; Saunders, R.A.; Pogson, A.N.; Hussmann, J.A.; Lenail, A.; Guna, A.; Mascibroda, L.; Wagner, E.J.; Adelman, K.; Lithwick-Yanai, G.; et al. Mapping information-rich genotype-phenotype landscapes with genome-scale Perturb-seq. *Cell* **2022**, *185*, 2559–2575.e28.
31. Jiang, Y.; Li, D.; Deng, H.; Ma, B.; Wang, X.; Wang, Q.; Yu, G. SoK: Agentic Skills – Beyond Tool Use in LLM Agents, 2026, [arXiv:cs.CR/2602.20867].
32. Meng, Q.; Wang, Y.; Chen, L.; Li, Y.; Wu, W.; Jiang, W.; Wang, Q.; Lu, C.; Gao, Y.; Wu, Y.; et al. Agent Harness for Large Language Model Agents: A Survey. *Preprints* **2026**. https://doi.org/10.20944/preprints202604.0428.v3.
33. Chen, M.; Wang, L.; Qu, B. Recursive Self-Improvement in AI: From Bounded Self-Refinement to Autonomous Research Loops. *arXiv:2607.07663* **2026**.
34. Chaloner, K.; Verdinelli, I. Bayesian experimental design: A review. *Statistical science* **1995**, pp. 273–304.
35. Rainforth, T.; Foster, A.; Ivanova, D.R.; Bickford Smith, F. Modern Bayesian experimental design. *Statistical Science* **2024**, *39*, 100–114.
36. Foster, A.; Ivanova, D.R.; Malik, I.; Rainforth, T. Deep Adaptive Design: Amortizing Sequential Bayesian Experimental Design. In Proceedings of the Proceedings of the 38th International Conference on Machine Learning. PMLR, 2021, Vol. 139, *Proceedings of Machine Learning Research*, pp. 3384–3395.
37. Lin, J.; Liu, S.; Pan, C.; Lin, L.; Dou, S.; Xi, Z.; Huang, X.; Yan, H.; Han, Z.; Gui, T.; et al. Agentic harness engineering: Observability-driven automatic evolution of coding-agent harnesses. *arXiv:2604.25850* **2026**.
38. Tang, X.; Jin, Q.; Zhu, K.; Yuan, T.; Zhang, Y.; Zhou, W.; Qu, M.; Zhao, Y.; Tang, J.; Zhang, Z.; et al. Risks of AI scientists: prioritizing safeguarding over autonomy. *Nature Communications* **2025**, *16*, 8317.

 doi:10.20944/preprints202608.2107.v1

39. Amershi, S.; Weld, D.; Vorvoreanu, M.; Fourney, A.; Nushi, B.; Collisson, P.; Suh, J.; Iqbal, S.; Bennett, P.N.; Inkpen, K.; et al. Guidelines for human-AI interaction. In Proceedings of the Proceedings of the 2019 chi conference on human factors in computing systems, 2019, pp. 1–13.
40. Everett, S.S.; Bunning, B.J.; Jain, P.; Lopez, I.; Agarwal, A.; Desai, M.; Gallo, R.; Goh, E.; Kadiyala, V.B.; Kanjee, Z.; et al. From tool to teammate in a randomized controlled trial of clinician-AI collaborative workflows for diagnosis. *NPJ Digital Medicine* **2026**, *9*, 409.
41. Cong, L.; Smerkous, D.; Wang, X.; Yin, D.; Zhang, Z.; Jin, R.; Wang, Y.; Gerasimiuk, M.; Dinesh, R.K.; Smerkous, A.; et al. LabOS: The AI-XR Co-Scientist That Sees and Works With Humans, 2025, [arXiv:cs.AI/2510.14861].
42. Stach, E.; DeCost, B.; Kusne, A.G.; Hattrick-Simpers, J.; Brown, K.A.; Reyes, K.G.; Schrier, J.; Billinge, S.; Buonassisi, T.; Foster, I.; et al. Autonomous experimentation systems for materials development: A community perspective. *Matter* **2021**, *4*, 2702–2726. https://doi.org/10.1016/j.matt.2021.06.036.
43. Leong, S.X.; Griesbach, C.E.; Zhang, R.; Darvish, K.; Zhao, Y.; Mandal, A.; Zou, Y.; Hao, H.; Bernales, V.; Aspuru-Guzik, A. Steering towards safe self-driving laboratories. *Nature Reviews Chemistry* **2025**, *9*, 707–722.
44. Zelikman, E.; Wu, Y.; Mu, J.; Goodman, N.D. STaR: Bootstrapping Reasoning With Reasoning. In Proceedings of the Advances in Neural Information Processing Systems, 2022, Vol. 35, pp. 15476–15488. https://doi.org/10.52202/068431-1126.
45. Hsieh, C.Y.; Li, C.L.; Yeh, C.k.; Nakhost, H.; Fujii, Y.; Ratner, A.; Krishna, R.; Lee, C.Y.; Pfister, T. Distilling Step-by-Step! Outperforming Larger Language Models with Less Training Data and Smaller Model Sizes. In Proceedings of the Findings of the Association for Computational Linguistics: ACL 2023. Association for Computational Linguistics, 2023, pp. 8003–8017. https://doi.org/10.18653/v1/2023.findings-acl.507.
46. Gururangan, S.; Marasović, A.; Swayamdipta, S.; Lo, K.; Beltagy, I.; Downey, D.; Smith, N.A. Don't Stop Pretraining: Adapt Language Models to Domains and Tasks. In Proceedings of the Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics. Association for Computational Linguistics, 2020, pp. 8342–8360. https://doi.org/10.18653/v1/2020.acl-main.740.
47. Mukherjee, S.; Mitra, A.; Jawahar, G.; Agarwal, S.; Palangi, H.; Awadallah, A. Orca: Progressive Learning from Complex Explanation Traces of GPT-4. *arXiv preprint arXiv:2306.02707* **2023**. https://doi.org/10.48550/arXiv.2306.02707.
48. Lightman, H.; Kosaraju, V.; Burda, Y.; Edwards, H.; Baker, B.; Lee, T.; Leike, J.; Schulman, J.; Sutskever, I.; Cobbe, K. Let's verify step by step. In Proceedings of the International Conference on Learning Representations, 2024, Vol. 2024, pp. 39578–39601.
49. Madaan, A.; Tandon, N.; Gupta, P.; Hallinan, S.; Gao, L.; Wiegreffe, S.; Alon, U.; Dziri, N.; Prabhumoye, S.; Yang, Y.; et al. Self-refine: Iterative refinement with self-feedback. *Advances in Neural Information Processing Systems* **2023**, *36*, 46534–46594.
50. Turpin, M.; Michael, J.; Perez, E.; Bowman, S.R. Language Models Don't Always Say What They Think: Unfaithful Explanations in Chain-of-Thought Prompting. In Proceedings of the Advances in Neural Information Processing Systems, 2023, Vol. 36, pp. 74952–74965. https://doi.org/10.52202/075280-3275.
51. Xiong, G.; Xie, E.; Williams, C.; Kim, M.; Shariatmadari, A.H.; Guo, S.; Bekiranov, S.; Zhang, A. Toward Reliable Scientific Hypothesis Generation: Evaluating Truthfulness and Hallucination in Large Language Models. In Proceedings of the Proceedings of the Thirty-Fourth International Joint Conference on Artificial Intelligence. International Joint Conferences on Artificial Intelligence Organization, 2025, pp. 7849–7857. https://doi.org/10.24963/ijcai.2025/873.
52. Shumailov, I.; Shumaylov, Z.; Zhao, Y.; Papernot, N.; Anderson, R.; Gal, Y. AI models collapse when trained on recursively generated data. *Nature* **2024**, *631*, 755–759.
53. Alemohammad, S.; Casco-Rodriguez, J.; Luzi, L.; Humayun, A.I.; Babaei, H.; LeJeune, D.; Siahkoohi, A.; Baraniuk, R. Self-consuming generative models go mad. In Proceedings of the International Conference on Learning Representations, 2024, Vol. 2024, pp. 53581–53608.
54. Uesato, J.; Kushman, N.; Kumar, R.; Song, F.; Siegel, N.; Wang, L.; Creswell, A.; Irving, G.; Higgins, I. Solving Math Word Problems with Process- and Outcome-Based Feedback. *arXiv preprint arXiv:2211.14275* **2022**. https://doi.org/10.48550/arXiv.2211.14275.
55. Phillips, L.; Martell, M.B.; Misra, A.; Stoisser, J.L.; Prada-Medina, C.A.; Donovan-Maiye, R.; Märtens, K. SynthPert: Enhancing LLM biological reasoning via synthetic reasoning traces for cellular perturbation prediction **2025**. [arXiv:cs.AI/2509.25346].
56. Amayuelas, A.; Sain, J.; Kaur, S.; Smiley, C. Grounding LLM reasoning with knowledge Graphs **2025**. [arXiv:cs.CL/2502.13247].

 doi:10.20944/preprints202608.2107.v1

57. Tan, X.W.; Tan, N.; Lee, G.; Kok, S. The Shape of Reasoning: Topological Analysis of Reasoning Traces in Large Language Models, 2025, [arXiv:2510.20665].
58. Chu, T.; Zhai, Y.; Yang, J.; Tong, S.; Xie, S.; Schuurmans, D.; Le, Q.V.; Levine, S.; Ma, Y. SFT Memorizes, RL Generalizes: A Comparative Study of Foundation Model Post-training. In Proceedings of the Proceedings of the 42nd International Conference on Machine Learning; Singh, A.; Fazel, M.; Hsu, D.; Lacoste-Julien, S.; Berkenkamp, F.; Maharaj, T.; Wagstaff, K.; Zhu, J., Eds. PMLR, 13–19 Jul 2025, Vol. 267, *Proceedings of Machine Learning Research*, pp. 10818–10838.
59. Rajani, N.; Gema, A.P.; Goldfarb-Tarrant, S.; Titov, I. Scalpel vs. Hammer: GRPO Amplifies Existing Capabilities, SFT Replaces Them. *arXiv preprint arXiv:2507.10616* **2025**. https://doi.org/10.48550/arXiv.2507.10616.
60. Jin, H.; Luan, S.; Ni, T.; Lyu, S.; Rabusseau, G.; Rabbany, R.; Precup, D.; Hamdaqa, M. RL Fine-Tuning Heals OOD Forgetting in SFT. *arXiv preprint arXiv:2509.12235* **2025**. https://doi.org/10.48550/arXiv.2509.12235.
61. Kang, F.; Kuchnik, M.; Padthe, K.; Vlastelica, M.; Jia, R.; Wu, C.J.; Ardalani, N. Quagmires in SFT–RL Post-Training: When High SFT Scores Mislead and What to Use Instead. *arXiv preprint arXiv:2510.01624* **2025**. https://doi.org/10.48550/arXiv.2510.01624.
62. Matsutani, K.; Takashiro, S.; Minegishi, G.; Kojima, T.; Iwasawa, Y.; Matsuo, Y. Rl squeezes, sft expands: A comparative study of reasoning llms. In Proceedings of the International Conference on Learning Representations, 2026, Vol. 2026, pp. 141482–141526.
63. Yue, Y.; Chen, Z.; Lu, R.; Zhao, A.; Wang, Z.; Song, S.; Huang, G. Does reinforcement learning really incentivize reasoning capacity in llms beyond the base model? *Advances in Neural Information Processing Systems* **2026**, *38*, 57654–57689.
64. Schulman, J.; Wolski, F.; Dhariwal, P.; Radford, A.; Klimov, O. Proximal Policy Optimization Algorithms, 2017. https://doi.org/10.48550/ARXIV.1707.06347.
65. Shao, Z.; Wang, P.; Zhu, Q.; Xu, R.; Song, J.; Bi, X.; Zhang, H.; Zhang, M.; Li, Y.K.; Wu, Y.; et al. DeepSeekMath: Pushing the Limits of Mathematical Reasoning in Open Language Models, 2024. https://doi.org/10.48550/ARXIV.2402.03300.
66. Yu, Q.; Zhang, Z.; Zhu, R.; Yuan, Y.; Zuo, X.; YuYue.; Dai, W.; Fan, T.; Liu, G.; Liu, J.; et al. DAPO: An Open-Source LLM Reinforcement Learning System at Scale. In Proceedings of the The Thirty-ninth Annual Conference on Neural Information Processing Systems, 2025.
67. Zheng, C.; Liu, S.; Li, M.; Chen, X.H.; Yu, B.; Gao, C.; Dang, K.; Liu, Y.; Men, R.; Yang, A.; et al. Group Sequence Policy Optimization, 2025, [arXiv:cs.LG/2507.18071].
68. Arjona-Medina, J.A.; Gillhofer, M.; Widrich, M.; Unterthiner, T.; Brandstetter, J.; Hochreiter, S. RUDDER: Return Decomposition for Delayed Rewards. In Proceedings of the Advances in Neural Information Processing Systems, 2019, Vol. 32, pp. 13544–13555.
69. Guez, A.; Silver, D.; Dayan, P. Efficient Bayes-Adaptive Reinforcement Learning Using Sample-Based Search. In Proceedings of the Advances in Neural Information Processing Systems, 2012, Vol. 25.
70. Kaelbling, L.P.; Littman, M.L.; Cassandra, A.R. Planning and Acting in Partially Observable Stochastic Domains. *Artificial Intelligence* **1998**, *101*, 99–134. https://doi.org/10.1016/S0004-3702(98)00023-X.
71. Silver, D.; Veness, J. Monte-Carlo Planning in Large POMDPs. In Proceedings of the Advances in Neural Information Processing Systems, 2010, Vol. 23.
72. Klissarov, M.; Bagaria, A.; Luo, Z.; Konidaris, G.; Precup, D.; Machado, M.C. Discovering temporal structure: An overview of hierarchical reinforcement learning **2025**. [arXiv:cs.AI/2506.14045].
73. Zhou, Y.; Zanette, A. ArCHer: training language model agents via hierarchical multi-turn RL. In Proceedings of the Proceedings of the 41st International Conference on Machine Learning. JMLR.org, 2024, ICML'24.
74. Messica, S.; Zhang, J.; Li, K.; Tsiligkaridis, T.; Zitnik, M. Adaptive Time Series Reasoning via Segment Selection. In Proceedings of the Forty-third International Conference on Machine Learning, 2026.
75. Moerland, T.M.; Broekens, J.; Plaat, A.; Jonker, C.M. Model-based reinforcement learning: A survey **2020**. [arXiv:cs.LG/2006.16712].
76. Zhang, L.; Yang, G.; Stadie, B.C. World model as a graph: Learning latent landmarks for planning. In Proceedings of the International Conference on Machine Learning. PMLR, 2021, pp. 12611–12620.
77. Zidan, A.H.; Pan, Y.; Jiang, H.; Yan, R.; Ruan, W.; Wu, Z.; Chen, L.; You, W.; Li, X.; Chen, B.; et al. World models: A comprehensive survey of architectures, methodologies, reasoning paradigms, and applications **2026**. [arXiv:cs.LG/2606.00133].
78. Roohani, Y.; Huang, K.; Leskovec, J. Predicting transcriptional outcomes of novel multigene perturbations with GEARS. *Nature Biotechnology* **2024**, *42*, 927–935.

 doi:10.20944/preprints202608.2107.v1

79. Gonzalez, G.; Lin, X.; Herath, I.; Veselkov, K.; Bronstein, M.; Zitnik, M. Combinatorial prediction of therapeutic perturbations using causally inspired neural networks. *Nature Biomedical Engineering* **2026**, *10*, 1008–1025. https://doi.org/10.1038/s41551-025-01481-x.
80. Panigrahi, S.S.; Videnović, J.; Brbic, M. HeurekaBench: A Benchmarking Framework for AI Co-scientist. In Proceedings of the The Fourteenth International Conference on Learning Representations, 2026.
81. Jansen, P.; Côté, M.A.; Khot, T.; Bransom, E.; Mishra, B.D.; Majumder, B.P.; Tafjord, O.; Clark, P. DiscoveryWorld: A Virtual Environment for Developing and Evaluating Automated Scientific Discovery Agents. In Proceedings of the The Thirty-eight Conference on Neural Information Processing Systems Datasets and Benchmarks Track, 2024.
82. Chen, Y.; Piękos, P.; Ostaszewski, M.; Laakom, F.; Schmidhuber, J. PhysGym: Benchmarking LLMs in Interactive Physics Discovery with Controlled Priors. In Proceedings of the The Thirty-ninth Annual Conference on Neural Information Processing Systems Datasets and Benchmarks Track, 2026.
83. Bazgir, A.; chandra Praneeth Madugula, R.; Zhang, Y. AgenticHypothesis: A Survey on Hypothesis Generation Using LLM Systems. In Proceedings of the Towards Agentic AI for Science: Hypothesis Generation, Comprehension, Quantification, and Validation, 2025.
84. Jacobsson, T.J. AI-generated hypotheses and the emergence of autonomous scientific discovery. *ACS Mater. Lett.* **2026**, *8*, 1457–1464.
85. Monteiro, L.M.O.; Chowdhury, N.B.; Oostrom, M.T.; McDermott, J.E.; Stratton, K.G.; Choudhury, S.; Bardhan, J.P. PathwaySeeker: Evidence-grounded AI reasoning over organism-specific metabolic networks. bioRxiv, 2026.
86. Zhao, H.; Ma, C.; Xu, F.; Kong, L.; Deng, Z.H. BioMaze: Benchmarking and enhancing large language models for biological pathway reasoning **2025**. [arXiv:cs.LG/2502.16660].
87. Yuan, W.; Pang, R.Y.; Cho, K.; Li, X.; Sukhbaatar, S.; Xu, J.; Weston, J. Self-rewarding language models. *ICML* **2024**.
88. Thornburg, Z.R.; Maytin, A.; Kwon, J.; Brier, T.A.; Gilbert, B.R.; Fu, E.; Gao, Y.L.; Quenneville, J.; Wu, T.; Li, H.; et al. Bringing the genetically minimal cell to life on a computer in 4D. *Cell* **2026**, *189*, 2582–2597.e27. https://doi.org/10.1016/j.cell.2026.02.009.
89. Shaw, D.E.; Maragakis, P.; Lindorff-Larsen, K.; Piana, S.; Dror, R.O.; Eastwood, M.P.; Bank, J.A.; Jumper, J.M.; Salmon, J.K.; Shan, Y.; et al. Atomic-Level Characterization of the Structural Dynamics of Proteins. *Science* **2010**, *330*, 341–346, [https://www.science.org/doi/pdf/10.1126/science.1187409]. https://doi.org/10.1126/science.1187409.
90. Jumper, J.; Evans, R.; Pritzel, A.; Green, T.; Figurnov, M.; Ronneberger, O.; Tunyasuvunakool, K.; Bates, R.; Žídek, A.; Potapenko, A.; et al. Highly accurate protein structure prediction with AlphaFold. *Nature* **2021**, *596*, 583–589. https://doi.org/10.1038/s41586-021-03819-2.
91. Trott, O.; Olson, A.J. AutoDock Vina: improving the speed and accuracy of docking with a new scoring function, efficient optimization, and multithreading. *Journal of computational chemistry* **2010**, *31*, 455–461.
92. Asai, A.; He, J.; Shao, R.; Shi, W.; Singh, A.; Chang, J.C.; Lo, K.; Soldaini, L.; Feldman, S.; D'Arcy, M.; et al. Synthesizing scientific literature with retrieval-augmented language models. *Nature* **2026**, *650*, 857–863. https://doi.org/10.1038/s41586-025-10072-4.
93. Perdigão, N.; Heinrich, J.; Stolte, C.; Sabir, K.S.; Buckley, M.J.; Tabor, B.; Signal, B.; Gloss, B.S.; Hammang, C.J.; Rost, B.; et al. Unexpected features of the dark proteome. *Proceedings of the National Academy of Sciences* **2015**, *112*, 15898–15903.
94. Duran-Frigola, M.; Pauls, E.; Guitart-Pla, O.; Bertoni, M.; Alcalde, V.; Amat, D.; Juan-Blanco, T.; Aloy, P. Extending the small-molecule similarity principle to all levels of biology with the Chemical Checker. *Nature Biotechnology* **2020**, *38*, 1087–1096.
95. Ahlmann-Eltze, C.; Huber, W.; Anders, S. Deep-learning-based gene perturbation effect prediction does not yet outperform simple linear baselines. *Nature Methods* **2025**, *22*, 1657–1661.
96. Wei, Z.; Wang, Y.; Gao, Y.; Wang, S.; Li, P.; Si, D.; Gao, Y.; Wu, S.; Li, D.; Dong, K.; et al. Benchmarking algorithms for generalizable single-cell perturbation response prediction. *Nature Methods* **2026**, *23*, 451–464. https://doi.org/10.1038/s41592-025-02980-0.
97. Palmer, G.; Du, S.; Politowicz, A.; Emory, J.P.; Yang, X.; Gautam, A.; Gupta, G.; Li, Z.; Jacobs, R.; Morgan, D. Calibration after bootstrap for accurate uncertainty quantification in regression models. *npj Computational Materials* **2022**, *8*, 115.
98. Tyralis, H.; Papacharalampous, G. A review of predictive uncertainty estimation with machine learning. *Artificial Intelligence Review* **2024**, *57*, 94.

 doi:10.20944/preprints202608.2107.v1

99. Merchant, A.; Batzner, S.; Schoenholz, S.S.; Aykol, M.; Cheon, G.; Cubuk, E.D. Scaling deep learning for materials discovery. *Nature* **2023**, *624*, 80–85. https://doi.org/10.1038/s41586-023-06735-9.
100. Cao, L.; Coventry, B.; Goreshnik, I.; Huang, B.; Sheffler, W.; Park, J.S.; Jude, K.M.; Marković, I.; Kadam, R.U.; Verschueren, K.H.G.; et al. Design of protein-binding proteins from the target structure alone. *Nature* **2022**, *605*, 551–560. https://doi.org/10.1038/s41586-022-04654-9.
101. Bennett, N.R.; Coventry, B.; Goreshnik, I.; Huang, B.; Allen, A.; Vafeados, D.; Peng, Y.P.; Dauparas, J.; Baek, M.; Stewart, L.; et al. Improving de novo protein binder design with deep learning. *Nature Communications* **2023**, *14*, 2625. https://doi.org/10.1038/s41467-023-38328-5.
102. Pacesa, M.; Nickel, L.; Schellhaas, C.; Schmidt, J.; Pyatova, E.; Kissling, L.; Barendse, P.; Choudhury, J.; Kapoor, S.; Alcaraz-Serna, A.; et al. One-shot design of functional protein binders with BindCraft. *Nature* **2025**, *646*, 483–492. https://doi.org/10.1038/s41586-025-09429-6.
103. Listov, D.; Goverde, C.A.; Correia, B.E.; Fleishman, S.J. Opportunities and challenges in design and optimization of protein function. *Nature Reviews Molecular Cell Biology* **2024**, *25*, 639–653.
104. Varga, J.K.; Ovchinnikov, S.; Schueler-Furman, O. actifpTM: a refined confidence metric of AlphaFold2 predictions involving flexible regions. *Bioinformatics* **2025**, *41*, btaf107, [https://academic.oup.com/bioinformatics/article-pdf/41/3/btaf107/62402836/btaf107.pdf]. https://doi.org/10.1093/bioinformatics/btaf107.
105. Dunbrack Jr, R.L. Rēs ipSAE loquunt: What's wrong with AlphaFold's ipTM score and how to fix it. *bioRxiv* **2025**.
106. Sabanza-Gil, V.; Barbano, R.; Pacheco Gutiérrez, D.; Luterbacher, J.S.; Hernández-Lobato, J.M.; Schwaller, P.; Roch, L. Best practices for multi-fidelity Bayesian optimization in materials and molecular research. *Nature Computational Science* **2025**, *5*, 572–581.
107. Trinh, T.H.; Wu, Y.; Le, Q.V.; He, H.; Luong, T. Solving olympiad geometry without human demonstrations. *Nature* **2024**, *625*, 476–482.
108. Wang, Z.; Li, Y.; Wu, Y.; Luo, L.; Hou, L.; Yu, H.; Shang, J. Multi-step problem solving through a verifier: An empirical analysis on model-induced process supervision. In Proceedings of the Findings of the Association for Computational Linguistics: EMNLP 2024, 2024, pp. 7309–7319.
109. Istrate, A.M.; Milletari, F.; Castrotorres, F.; Tomczak, J.M.; Torkar, M.; Li, D.; Karaletsos, T. rbio1-training scientific reasoning LLMs with biological world models as soft verifiers. *bioRxiv* **2025**, pp. 2025–08.
110. Gao, L.; Schulman, J.; Hilton, J. Scaling laws for reward model overoptimization. In Proceedings of the International Conference on Machine Learning. PMLR, 2023, pp. 10835–10866.
111. Rettie, S.A.; Juergens, D.; Adebomi, V.; Bueso, Y.F.; Zhao, Q.; Leveille, A.N.; Liu, A.; Bera, A.K.; Wilms, J.A.; Üffing, A.; et al. Accurate de novo design of high-affinity protein-binding macrocycles using deep learning. *Nature Chemical Biology* **2025**, *21*, 1948–1956. https://doi.org/10.1038/s41589-025-01929-w.
112. Bates, T.A.; Gurmessa, S.K.; Weinstein, J.B.; Trank-Greene, M.; Wrynla, X.H.; Anastas, A.; Anley, T.W.; Hinchliff, A.; Shinde, U.; Burke, J.E.; et al. Biolayer interferometry for measuring the kinetics of protein–protein interactions and nanobody binding. *Nature Protocols* **2025**, *20*, 861–883. https://doi.org/10.1038/s41596-024-01079-8.
113. MacLeod, B.P.; Parlane, F.G.; Morrissey, T.D.; Häse, F.; Roch, L.M.; Dettelbach, K.E.; Moreira, R.; Yunker, L.P.; Rooney, M.B.; Deeth, J.R.; et al. Self-driving laboratory for accelerated discovery of thin-film materials. *Science Advances* **2020**, *6*, eaaz8867.
114. Baker, M. Reproducibility crisis: Blame it on the antibodies. *Nature* **2015**, *521*, 274.
115. Baker, M. 1,500 scientists lift the lid on reproducibility. *Nature* **2016**, *533*, 452–454. https://doi.org/10.1038/533452a.
116. Zhang, J.; Cammarata, L.; Squires, C.; Sapsis, T.P.; Uhler, C. Active learning for optimal intervention design in causal models. *Nature Machine Intelligence* **2023**, *5*, 1066–1075. https://doi.org/10.1038/s42256-023-00719-0.
117. Tosh, C.; Tec, M.; White, J.B.; Quinn, J.F.; Ibanez Sanchez, G.; Calder, P.; Kung, A.L.; Dela Cruz, F.S.; Tansey, W. A Bayesian active learning platform for scalable combination drug screens. *Nature Communications* **2025**, *16*, 156. https://doi.org/10.1038/s41467-024-55287-7.
118. Rubbi, A.; Merchant, A.; Ogden, S.; Akbarnejad, A.; Liò, P.; Vakili, S.; Lotfollahi, M. Many Needles in a Haystack: Active Hit Discovery for Perturbation Experiments. *International Conference on Machine Learning* **2026**.
119. Rich, R.L.; Papalia, G.A.; Flynn, P.J.; Furneisen, J.; Quinn, J.; Klein, J.S.; Katsamba, P.S.; Waddell, M.B.; Scott, M.; Thompson, J.; et al. A global benchmark study using affinity-based biosensors. *Analytical biochemistry* **2009**, *386*, 194–216.

 doi:10.20944/preprints202608.2107.v1

120. Bennett, N.R.; Watson, J.L.; Ragotte, R.J.; Borst, A.J.; See, D.L.; Weidle, C.; Biswas, R.; Yu, Y.; Shrock, E.L.; Ault, R.; et al. Atomically accurate de novo design of antibodies with RFdiffusion. *Nature* **2026**, *649*, 183–193. https://doi.org/10.1038/s41586-025-09721-5.
121. Lindley, D.V. On a measure of the information provided by an experiment. *The Annals of Mathematical Statistics* **1956**, *27*, 986–1005.
122. Neiswanger, W.; Ramdas, A. Uncertainty quantification using martingales for misspecified Gaussian processes. In Proceedings of the Algorithmic learning theory. PMLR, 2021, pp. 963–982.
123. Folch, J.P.; Lee, R.M.; Shafei, B.; Walz, D.; Tsay, C.; Van Der Wilk, M.; Misener, R. Combining multi-fidelity modelling and asynchronous batch Bayesian Optimization. *Computers & Chemical Engineering* **2023**, *172*, 108194.
124. Abolhasani, M.; Kumacheva, E. The rise of self-driving labs in chemical and materials sciences. *Nature Synthesis* **2023**, *2*, 483–492.
125. Burger, B.; Maffettone, P.M.; Gusev, V.V.; Aitchison, C.M.; Bai, Y.; Wang, X.; Li, X.; Alston, B.M.; Li, B.; Clowes, R.; et al. A mobile robotic chemist. *Nature* **2020**, *583*, 237–241. https://doi.org/10.1038/s41586-020-2442-2.
126. Yang, J.; Lal, R.G.; Bowden, J.C.; Astudillo, R.; Hameedi, M.A.; Kaur, S.; Hill, M.; Yue, Y.; Arnold, F.H. Active learning-assisted directed evolution. *Nature Communications* **2025**, *16*, 714.
127. Zhang, Q.; Chen, W.; Qin, M.; Wang, Y.; Pu, Z.; Ding, K.; Liu, Y.; Zhang, Q.; Li, D.; Li, X.; et al. Integrating protein language models and automatic biofoundry for enhanced protein evolution. *Nature Communications* **2025**, *16*, 1553. https://doi.org/10.1038/s41467-025-56751-8.
128. Frey, N.C.; Hötzel, I.; Stanton, S.D.; Kelly, R.; Alberstein, R.G.; Makowski, E.K.; Martinkus, K.; Berenberg, D.; Bevers, J.; Bryson, T.; et al. Lab-in-the-loop therapeutic antibody design with deep learning. *bioRxiv* **2025**, [https://www.biorxiv.org/content/early/2025/11/08/2025.02.19.639050.full.pdf]. https://doi.org/10.1101/2025.02.19.639050.
129. McDonald, M.A.; Koscher, B.A.; Canty, R.B.; Zhang, J.; Ning, A.; Jensen, K.F. Bayesian optimization over multiple experimental fidelities accelerates automated discovery of drug molecules. *ACS central science* **2025**, *11*, 346–356.
130. Kennedy, M.C.; O'Hagan, A. Predicting the output from a complex computer code when fast approximations are available. *Biometrika* **2000**, *87*, 1–13.
131. Mikkola, P.; Martinelli, J.; Filstroff, L.; Kaski, S. Multi-fidelity Bayesian optimization with unreliable information sources. In Proceedings of the International Conference on Artificial Intelligence and Statistics. PMLR, 2023, pp. 7425–7454.
132. Mitchener, L.; Laurent, J.M.; Andonian, A.; Tenmann, B.; Narayanan, S.; Wellawatte, G.P.; White, A.; Sani, L.; Rodriques, S.G. BixBench: a comprehensive benchmark for llm-based agents in computational biology. *arXiv:2503.00096* **2025**.
133. Nair, S.; Gunsalus, L.; Orcutt-Jahns, B.; Rossen, J.; Lal, A.; Donno, C.D.; Celik, M.H.; Fletez-Brant, K.; Xie, X.; Bravo, H.C.; et al. Agentic systems are adept at solving well-scoped, verifiable problems in computational biology. *bioRxiv* **2026**, pp. 2026–04.
134. Sun, Y.; Han, X.; Zhang, W.; Pang, Y.; Wang, T.; Cao, Y.; Huang, Y.; Duroiu, C.; Zhang, H.; Lin, J.; et al. Agents' Last Exam. *arXiv:2606.05405* **2026**.
135. Gao, S.; Su, Y.; Sui, P.; Ginder, C.; Zitnik, M. Qworld: Question-Specific Evaluation Criteria for LLMs. *arXiv:2603.23522* **2026**.
136. Ektefaie, Y.; Shen, A.; Bykova, D.; Marin, M.G.; Zitnik, M.; Farhat, M. Evaluating generalizability of artificial intelligence models for molecular datasets. *Nature Machine Intelligence* **2024**, *6*, 1512–1524.
137. Mialon, G.; Fourrier, C.; Wolf, T.; LeCun, Y.; Scialom, T. Gaia: a benchmark for general ai assistants. In Proceedings of the International Conference on Learning Representations, 2024, Vol. 2024, pp. 9025–9049.
138. Zhao, W.; Queralta, J.P.; Westerlund, T. Sim-to-Real Transfer in Deep Reinforcement Learning for Robotics: a Survey. *CoRR* **2020**, *abs/2009.13303*, [2009.13303].
139. de la Fuente, J.; Serrano, G.; Veleiro, U.; Casals, M.; Vera, L.; Pizurica, M.; Gómez-Cebrián, N.; Puchades-Carrasco, L.; Pineda-Lucena, A.; Ochoa, I.; et al. Towards a more inductive world for drug repurposing approaches. *Nature Machine Intelligence* **2025**, *7*, 495–508. https://doi.org/10.1038/s42256-025-00987-y.
140. Graber, D.; Stockinger, P.; Meyer, F.; Mishra, S.; Horn, C.; Buller, R. Resolving data bias improves generalization in binding affinity prediction. *Nature Machine Intelligence* **2025**, *7*, 1713–1725. https://doi.org/10.1038/s42256-025-01124-5.
141. Drazen, J.M.; Haug, C.J. Trials of AI Interventions Must Be Preregistered. *NEJM AI* **2024**, *1*, AIe2400146, [https://ai.nejm.org/doi/pdf/10.1056/AIe2400146]. https://doi.org/10.1056/AIe2400146.

 doi:10.20944/preprints202608.2107.v1

142. Lakens, D.; Mesquida, C.; Rasti, S.; Ditroilo, M. The benefits of preregistration and Registered Reports. *Evidence-Based Toxicology* **2024**, *2*, 2376046.
143. Gulcehre, C.; Paine, T.L.; Srinivasan, S.; Konyushkova, K.; Weerts, L.; Sharma, A.; Siddhant, A.; Ahern, A.; Wang, M.; Gu, C.; et al. Reinforced self-training (ReST) for language modeling. *arXiv:2308.08998* **2023**.
144. Macarron, R.; Banks, M.N.; Bojanic, D.; Burns, D.J.; Cirovic, D.A.; Garyantes, T.; Green, D.V.S.; Hertzberg, R.P.; Janzen, W.P.; Paslay, J.W.; et al. Impact of high-throughput screening in biomedical research. *Nature Reviews Drug Discovery* **2011**, *10*, 188–195. https://doi.org/10.1038/nrd3368.
145. Deng, C.; Zhao, Y.; Tang, X.; Gerstein, M.; Cohan, A. Investigating data contamination in modern benchmarks for large language models. In Proceedings of the Proceedings of the 2024 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 1: Long Papers), 2024, pp. 8706–8719.
146. Blum, A.; Hardt, M. The ladder: A reliable leaderboard for machine learning competitions. In Proceedings of the International Conference on Machine Learning. PMLR, 2015, pp. 1006–1014.
147. Stanford Institute for Human-Centered Artificial Intelligence. Technical Performance https://hai.stanford.edu/ai-index/2025-ai-index-report/technical-performance (2026).
148. Akhtar, M.; Reuel, A.; Soni, P.; Ahuja, S.; Ammanamanchi, P.S.; Rawal, R.; Zouhar, V.; Yadav, S.; Whitehouse, C.; Ki, D.; et al. When AI Benchmarks Plateau: A Systematic Study of Benchmark Saturation. In Proceedings of the Proceedings of the 43rd International Conference on Machine Learning. PMLR, 2026, Vol. 306, *Proceedings of Machine Learning Research*.
149. Gao, S.; Zhu, R.; Sui, P.; Kong, Z.; Aldogom, S.; Huang, Y.; Noori, A.; Shamji, R.; Parvataneni, K.; Tsiligkaridis, T.; et al. Democratizing AI scientists using ToolUniverse. *arXiv:2509.23426* **2025**.
150. Ide, E.; Talamas, E. Artificial intelligence in the knowledge economy. *Journal of Political Economy* **2025**, *133*, 3762–3800.