RESEARCH ARTICLE | BENCHMARK DESIGN AND REFERENCE IMPLEMENTATION

A Digital Twin for Every PCL Node

Evaluating scientific agents from instrument commands to defensible measurements

PCL-Bench project Working manuscript · November 5, 2026

Abstract. A scientific agent can select the right sample and still report the wrong result. A lost calibration read, an uncorrected instrument offset, or a failed reference can alter the scientific conclusion after the experimental design has been chosen. We introduce PCL-Bench, an executable benchmark for instrument-level decisions across the US National Science Foundation's Programmable Cloud Laboratories (PCL) Test Bed. We map the public research programs of all 20 teams to candidate tasks, and release three synthetic characterization environments with a common action interface. In a paired software experiment across 20 seeds per task, a reference policy completes all 120 clean and fault-injected episodes. Removing recovery eliminates all 60 successes in the fault condition. A policy that acquires every reading but implements a deliberately incorrect analysis fails all 120 episodes. As a separate test of reuse, we replay four published chemistry cases. Over 160 campaigns on measured outcomes, a local search rule reduces mean terminal yield regret from 9.90 to 7.55 percentage points compared with uniform sampling. We specify the additional trace calibration, expert review and held-out evaluation needed before a node-level score can support a claim about real laboratory performance.

1 Introduction

Imagine that an agent requests a fluorescence reading from a blank, a standard and an unknown protein sample. The blank succeeds. The tray faults on the standard. If the agent simply retries, it exhausts its budget. If it restores the instrument, obtains all three values and forgets to subtract the blank, it can submit a plausible concentration with the wrong value. An evaluation that records only whether the instrument ran will miss this distinction.

The NSF PCL Test Bed brings this problem into focus. Its 20 awardees span biology, chemistry, materials characterization and electronics [1, 2]. At Northwestern, DREAM's protein engineering program includes binding, enzyme kinetics, folding and stability [3]. Other centers study thin films, diffraction and manufacturing. These programs share questions about controls, instrument state and measurement uncertainty, even when their scientific endpoints have different units.

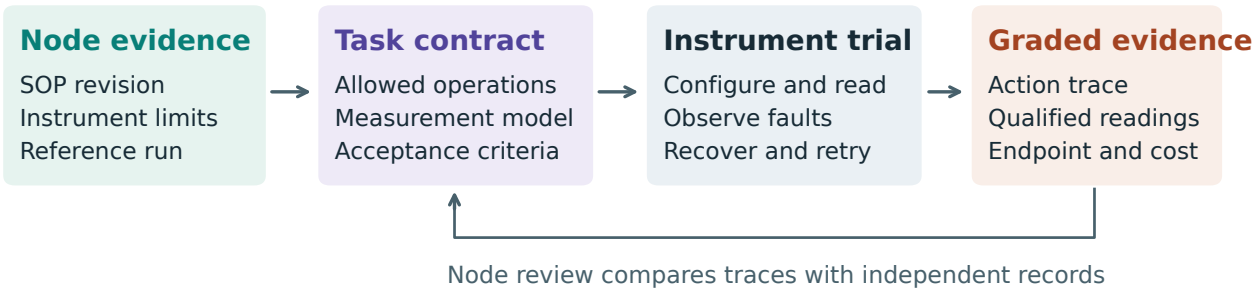


Figure 1: Proposed route from node evidence to a graded agent trial. The release implements the task contract, trial and grader for synthetic fixtures. A collaborator's reference records and review would establish fidelity for a specific site and operating range.

2 What prior work already measures

Autonomous Science Bench is the closest network-level precedent. Its initial release evaluates six biology, chemistry and materials campaigns, grading measured discoveries and predictions on unmeasured designs. Its roadmap explicitly includes digital twins [4]. One could attach an execution trial to a campaign decision: a request for one measurement produces an instrument trace, a qualified observation and a cost. Replacing an idealized lookup with that trial would let researchers ask whether a campaign still works when some readouts are invalid or delayed. Building and validating this bridge is a future experiment.

Prior object of evaluation	**What the source already provides**	**Candidate extension for PCL**
Protocol reasoning [5–9]	Detecting procedural errors, planning steps and interpreting research artifacts.	Make the consequence of a missing control observable at readout time.
Protocol compilation [10–12]	Converting descriptions into structured steps or executable procedures and checking their execution.	Add a postcondition on measured units, calibration and uncertainty to a compiler test.
Instrument interfaces [13–15]	GUI operation, microscopy and stateful virtual measurements. q-devBench includes a clean-data analysis ablation.	Run the same science objective against two device contracts and local references.
Embodied operation [16–19]	Digital lab assets, physical manipulation, perception and video-based reconstruction.	Connect a supported physical event such as a failed load to the data record it invalidates.
Safety and state [20–24]	Trajectory-level checks, safety reasoning or stateful execution in simulation.	Distinguish an attempted blocked operation from a recovery and from a missing reading.
Discovery and optimization [4, 25–30]	Interactive experimentation, feedback and choosing the next experiment.	Charge for invalid observations and compare follow-up choices at matched remaining budgets.
Characterization [31–33]	Scientific endpoints for spectra, diffraction and protein fitness.	Pair an endpoint with the agent's acquisition settings and quality-control history.

Table 1: Comparison by evaluated capability. The proposed extensions describe new task designs. The source catalog records one concrete adaptation for every work.

Physical demonstrations such as Coscientist [34] and autonomous cell-free protein synthesis [35] show how real hardware and resource costs change the problem. Science's Last Exam documents scientific workflow tasks across modalities [36]. Our synthesis uses the distinct evaluation units in these works to design an interface between instrument execution, quantitative analysis and campaign decisions. We do not report a cross-benchmark leaderboard because scores collected under those units have different meanings.

3 An executable contract for characterization

For node n and protocol family p, define a contract $C_{np} = (A, O, T, M, B, G, E)$. A specifies tool names, arguments, units and preconditions. O defines what the agent can observe. T updates instrument and sample state, while M generates a measurement conditional on that state. B is a finite budget, G is a grading function, and E identifies evidence for the assumptions. The separation of T and M matters: a tray can return to a ready state without producing the missing reading. A successful recovery must be followed by a fresh acquisition.

Released interface. The Python environment and a line-delimited JSON subprocess accept four operations: `configure`, `measure`, `recover` and `submit`. The agent observes operating bounds, valid acquisition points, the documented recovery operation, recorded readings and remaining actions. The seeded target is hidden from the observation. Every attempt, including a malformed action, consumes one of 15 steps. An out-of-range numeric configuration increments the interlock counter. The three fixtures expose a single supported recovery, which makes them suitable for testing the grader but easy to solve with a task-specific policy.

Grading. Let R_p denote required acquisition points, D the set of distinct points successfully read, v the number of interlock violations, $\hat{y}$ the submitted estimate and y its hidden target. The released success predicate is

$$S = \mathbf{1}\{R_p \subseteq D\}\, \mathbf{1}\{v = 0\}\, \mathbf{1}\{\hat{y} \text{ is finite and } |\hat{y} - y| \le \epsilon_p\}.$$

Timeouts have no numerical estimate. We publish their count separately and compute mean absolute error only on submitted estimates. This avoids treating a policy that never reports a result as numerically accurate.

Capability	Observable in the current release	Evidence needed for a node task
Instrument and API use	Legal settings, coordinates and tool return status.	Versioned device manual or driver test.
Planning and orchestration	Ordering of configuration, acquisition and submission on one device.	Sample lineage across instruments, resource locks and recorded handoffs.
Safety	Interlock on out-of-range settings.	Owner-defined prohibited actions and safe escalation.
Failure recovery	One visible, recoverable fault in the injected condition.	Anonymized fault traces and independent recovery examples.
Quality control	Required read coverage including specified controls.	Control acceptance ranges, drift or replicate evidence and rejection criteria.
Analysis and iteration	A quantitative endpoint in physical units.	Calibrated uncertainty and a next-experiment decision under budget.

Table 2: Six capability axes. The starter suite supports one-instrument episodes. Multi-instrument scheduling, richer QC and iterative science require separate validation.

Interoperability. LabOP represents protocol primitives and control flow [37]. SiLA 2 describes instrument features [38]. The Opentrons simulator provides protocol-level execution checks [39], while the Model Hardware Standard preview describes agent-facing device interfaces [40]. A node adapter can use these representations as evidence for operations. None by itself establishes an assay's response function. The current package uses a native JSON contract.

4 Paired evaluation of the reference implementation

The software experiment covers three deterministic task families, each with 20 parameter seeds, two conditions and three policies ($3 \times 20 \times 2 \times 3 = 360$ episodes). Clean and fault-injected episodes with the same task and seed share ground truth. In the fault condition the first acquisition at a seed-selected point fails once, the instrument blocks further reads, and the documented recovery clears it. The tests use no API models or real instruments.

Fluorescence calibration. A virtual plate reader returns intensities for a blank b, a 10 μM standard s and an unknown u. The endpoint is $10(u-b)/(s-b)$, accepted within 0.05 μM. The synthetic response is linear and noiseless. The shortcut policy uses $10u/s$, making background signal contribute to its estimate.

Absorbance peak. A virtual spectrophotometer returns readings at 440 to 540 nm in 20 nm steps. The hidden peak lies on the grid. An estimate within 0.5 nm passes. The shortcut policy returns the location of a minimum. This deliberate sign error demonstrates that full spectral coverage alone is insufficient.

I–V resistance. A virtual source meter supplies voltage at -1, 0 and $+1$ mA. A seed-specific contact-voltage offset enters all readings. The reference estimate is $[V(+1) - V(-1)]/(0.002\,\mathrm{A})$, with a 1 Ω tolerance. The shortcut estimates resistance from the positive reading alone.

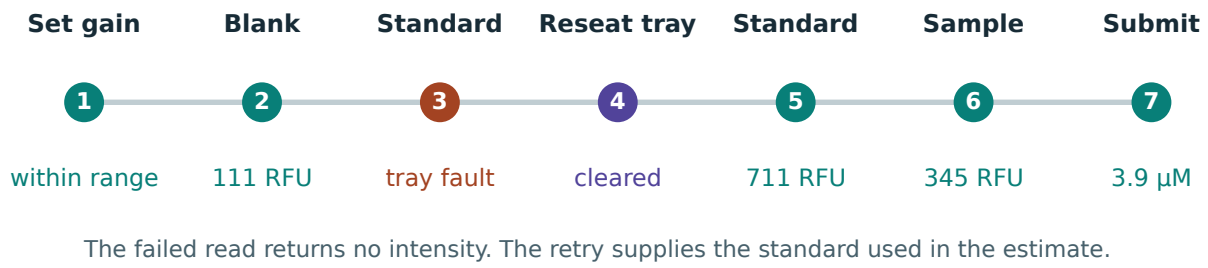


Figure 2: A complete reference episode replayed from the released JSON trace (fluorescence, seed 7). The instrument blocks acquisition of the standard until the tray is reseated. The values shown are synthetic.

Controls. The reference policy configures inside the reported limits, acquires the required points, recovers when prompted and computes the correct endpoint. The no-recovery policy removes only the fault-handling branch. The analysis-shortcut policy performs the same acquisitions and recoveries as the reference, then applies the family-specific incorrect estimator. These manipulations isolate two failure mechanisms by construction.

Reproducibility. Each row in `results/scripted-controls.json` records the task, seed, condition, policy, outcome and diagnostics. The aggregate table and the full seed-7 trace are included. An external agent can drive the same observation/action protocol. For a blinded evaluation, isolate the harness and hide the fixtures, seeds and grader from the model. Public test cases are appropriate for development.

5 Results and interpretation

The reference policy passes 60/60 clean and 60/60 injected episodes. Without recovery, the same policy passes every clean case and times out on every injected case. Correct recovery costs two additional actions per episode: the failed acquisition and the recovery action. The analysis shortcut completes every required read and recovers from all 60 injected faults, yet passes no endpoint in either condition. No policy in this matrix attempts an out-of-bounds configuration. A separate unit test exercises the interlock.

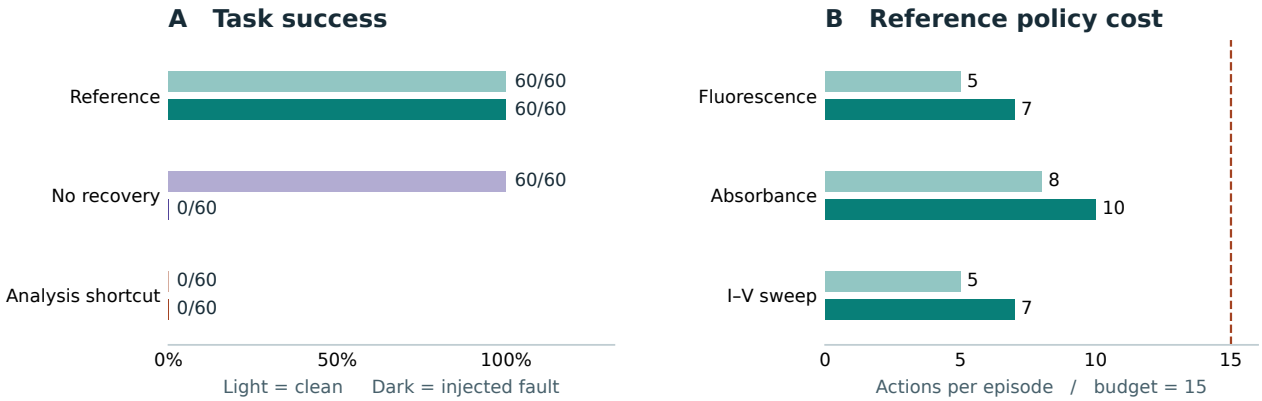


Figure 3: Panel A pools three families with 20 runs each per policy and condition. Panel B shows the action cost separately for each family. All runs and group means come from the published JSON output.

Task family	Unit	Pass tolerance	Shortcut mean absolute error
Fluorescence concentration	μM	0.05	0.592
Grid peak wavelength	nm	0.5	70.0
Resistance from I–V sweep	Ω	1.0	10.221

Table 3: Endpoint errors by family, computed over 20 estimates per condition. Each shortcut's estimate is identical across the matched clean and recovered conditions. The no-recovery control submitted zero estimates under fault.

What these results establish. They verify a property of the software: keeping the endpoint fixed while changing the execution path can change success, and keeping the readings fixed while changing the estimator can also change success. Ground truth and response functions are supplied by the simulator. The paired seeds are fixtures from three simple programs, so an interval estimated across them would say little about physical laboratories or the population of scientific tasks.

Limitations. Settings constrain access but do not continuously change the synthetic response. Inputs are noiseless, faults are visible, and the recovery name is exposed. The benchmark does not currently test perception on a real console, diagnose a latent hardware fault, fit a general physical model, or evaluate a learned agent. Real controls can disagree without yielding a unique diagnosis. For such cases, abstention or a discriminating follow-up experiment must receive credit if a node expert judges it appropriate.

6 Replay of published reaction measurements

To test transfer from an existing benchmark with measured outcomes, we implemented a separate replay of four Suzuki–Miyaura optimization cases. Reizman and colleagues collected reaction outcomes under automated feedback [49]. We use the historical tables distributed by Summit [46]. The adapter fetches four files at a fixed upstream Git commit and verifies their SHA-256 hashes. It reproduces the study without redistributing the upstream rows.

Task. Each candidate is a tuple of catalyst, residence time, temperature and catalyst loading. Within a case, we average yields across exact duplicate settings. The four tables contain 96, 96, 96 and 97 measured records, yielding 81, 80, 93 and 95 unique settings, respectively. A policy may query ten distinct settings and observe only the averaged yields it requests. Unmeasured conditions are outside the action space. Define terminal simple regret as

$$r_{10} = \max_{x \in X} \bar{y}(x) - \max_{x \in Q_{10}} \bar{y}(x),$$

where X is the measured candidate pool, Q_{10} is the set queried, and $\bar{y}$ is yield in percentage points.

Paired design. Uniform sampling and a local greedy policy receive the same first three settings for each seed. The latter then queries the unmeasured point closest to the best observed setting. Distance is the sum of a catalyst mismatch indicator and range-normalized absolute differences in the three continuous variables. We ran 20 paired seeds per case, for 160 campaigns. The sampling unit here is a campaign in a fixed historical pool. These runs are reported separately from the instrument controls.

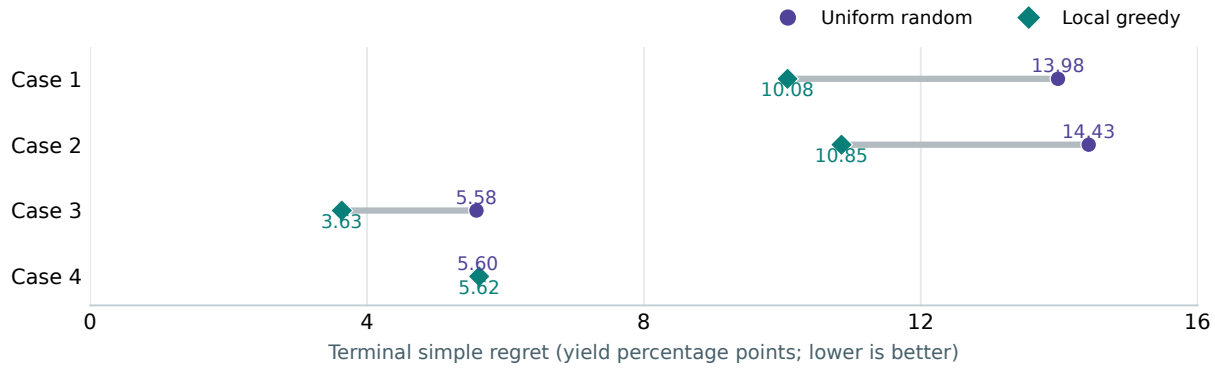


Figure 4: Mean terminal regret over 20 seeds for each of four historical reaction cases. Labels show percentage points of yield. Cases 1–3 favor the local rule at this budget. In case 4 the means differ by only 0.02 percentage points.

Results. Averaging the four case means with equal weight gives 9.90 percentage points of regret for uniform sampling and 7.55 for local greedy. The greatest reduction occurs in case 1 (13.98 to 10.08); case 4 shows a small reversal (5.60 to 5.62). A local search heuristic can exploit the geometry of some recorded pools while offering little advantage in others. The comparison does not estimate performance outside the measured settings or on an independent prospective campaign. The historical record does not provide node instrument commands, failed physical runs or grades for recovery. Those require additional local evidence.

7 Building a benchmark across the 20 programs

The accompanying mapping connects each of the 20 NSF award records to one proposed task and a candidate grading rule (Appendix A). Across the roster, 45 links connect these candidate tasks to 21 previous works. For each link the crosswalk identifies what to retain, what to adapt, the minimum real records to request and a diagnostic to hold out. It identifies specific starting points for discussion. An award's scientific scope supports the choice of problem area. Local operating conditions, instruments, data permission and acceptance thresholds require input from the node. No external node review or data contribution is claimed here.

From an existing benchmark to a node-specific case. Start with a reusable capability and identify the new supervision it needs. For example, a diffraction identification task may score the correct structure from a clean pattern [32]. An ELECTRA case could instead present two incomplete acquisitions and a remaining exposure budget. An expert would identify admissible follow-up measurements and decide when the evidence is insufficient. The adapter would return the selected pattern with its exposure and provenance. A grader would check the physical budget, evidence quality, and final interpretation. Comparing the final answer on a clean reference pattern with the closed-loop result would expose acquisition-induced error. Different diffraction modalities require their own signal and noise models.

Evidence ladder. A proposed case is a public design idea. A synthetic fixture adds executable semantics and unit tests. A replay case requires permission to use a historical trace plus a split manifest. A calibrated simulator must be tested against independent records with predeclared tolerances on readouts, event order and fault behavior. A node-reviewed case also needs local review of bounds, controls, alternatives and scoring. Each release should publish the highest level actually achieved and the date and scope of review. Synthetic parameter seeds do not advance a case up this ladder.

Counts that preserve scientific diversity. The proposed six-month target is 10,000 distinct scorable task instances and 100,000 agent episodes. Record the node, source protocol and revision, scientific objective, device profile, perturbation, split, provenance and grader version. Deduplicate by protocol lineage and by normalized instruction and action graphs. Count a paraphrase under the same underlying instance. Sampling a new campaign seed adds an episode unless the task contract and its decision actually change. Report accepted base protocols, families, instances and episodes separately; quotas should follow contributed workflows and local validation capacity.

Held-out tests. Divide by protocol lineage before generating paraphrases, variants or traces. Reserve unseen instrument configurations, then entire sites and scientific families once participation supports these splits. Compare a documented reference, a missing-recovery control, a data-analysis-only control on the oracle readouts, and candidate model agents on matched budgets. Record per-node macro success, invalid-action rate, attempt and recovery counts, endpoint error in local units, abstention quality and cost. Resample independent protocol families within sites for uncertainty intervals, and publish subgroup sample sizes. With only one contributing site, cross-site generalization remains unmeasured.

Availability and stewardship. The repository includes code, raw pilot results, an example adapter, the node map, the literature index and manuscript sources. Links to prior work identify material the reader can inspect. Importing third-party tasks, vendor manuals or site traces is a separate step with an explicit license or permission record. A node may retain private source records and release a validator or approved aggregate results. No new physical experiment is needed to run this reference release.

A Twenty node-specific starting points

Each row links directly to the NSF award. The right column is our proposed task. The JSON profile supplies a suggested grading observable and a stable award ID for each row. These are questions for collaborators to validate against their own workflows. The names also follow the PCL kickoff program [2].

NSF node and award	Candidate task for review
DREAM Northwestern University #2607522	Estimate a biosensor response curve after excluding a plate with failed controls.
CLAIRE Boston University #2607506	Compare cell-free reporter designs across plates with different reader gains.
CRAB Lab University of Maryland #2607564	Detect drift in a bioprocess sensor before changing a simulated feed schedule.
AI-BIOME City College of New York #2607601	Choose a repeat characterization that distinguishes two candidate assemblies.
AI Science Foundry Carnegie Mellon University #2607594	Route an assay to a compatible instrument when its scheduled reader is unavailable.
SPEED NC State University #2607495	Pick the next solution-phase experiment after a reference spectrum drifts.
COAST UC Santa Barbara #2607521	Compare viscoelastic curves for candidate soft materials after a geometry change.
ELECTRA Caltech #2607629	Select a follow-up diffraction measurement from incomplete or degraded patterns.
GEMSTONE Stanford University #2607573	Bind an abstract transfer and readout protocol to two different mock device APIs.
Reaction Discovery Scripps Research #2607628	Rank candidate conditions after detecting a missing internal standard.

Example of a shared scoring concept. A lost reference can invalidate a biological plate or a chemical spectrum. What transfers is the requirement for traceable calibration. The quantity used for calibration, its acceptance interval and the action needed to recover must come from the respective laboratory.

APPENDIX A / NODE TASK MAP, CONTINUED

NSF node and award	Candidate task for review
PoLARIS University of Chicago #2607613	Choose the next polymer composition after a replicated property disagrees.
PCL-AIMMPF Georgia Tech #2607581	Compare manufacturing settings using matched reference coupons.
AIMD-NET Johns Hopkins University #2607526	Route alloy samples for follow-up tests after uncertainty in a property estimate.
ATHENA University of Tennessee, Knoxville #2607469	Reacquire a scan region after simulated tip drift degrades an image.
ICoN-PCL Purdue University #2607533	Select a follow-up probe to distinguish two candidate material states.
Self-Driving Nanofab Virginia Tech #2607580	Reschedule a fabrication and metrology queue after a tool becomes unavailable.
LATTICE Penn State #2607536	Identify a thickness gradient from spatially sparse metrology and choose new sites.
Electronics Prototyping ADOM Industries #2607577	Estimate component resistance from a simulated sweep with a bad contact.
READINESS Rice University #2607553	Choose a follow-up characterization after a candidate film misses its target.
AURORA University of Utah #2607535	Decide whether a stacked flake is aligned well enough for transport measurement.

From award scope to data. For each row, record who can release a reference run, which fields are restricted, which physical units the instrument reports and how the local team adjudicates borderline cases. Until these questions are resolved, the row remains a proposed mapping.

B Source selection and adaptation

We used a targeted, non-systematic search through November 5, 2026. Queries combined laboratory agents, scientific instrument control, protocol compilation, digital twins, characterization, experimental design and autonomous laboratory benchmarks. We followed references from close papers and checked primary author, publisher, project and standards pages. Inclusion required a concrete task, evaluation method, protocol representation, instrument interface, or physical system that could inform a PCL case. Different evaluations with similar names were retained separately. The catalog is extensible and has no claim to exhaust all fields.

Evidence from physical systems. A mobile robotic chemist [41] and A-Lab's solid-state synthesis system [42] provide precedents for closing the loop around instruments. The latter article has a publisher correction, linked in our catalog. Measurement claims require an independent reference for the particular material or assay. Illinois' iCloudBiofoundry [43] describes how biological objectives can reach device-level automation with digital-twin checks. The BioFaudry asset collection [44] offers an example of digitized lab geometry. Its data card is public and asset access is gated. A geometrically accurate reader would still need a separate model of saturation, controls and noise before it could produce a valid simulated assay.

Reuse at the correct level. ChemWorld provides a close precedent for replayable chemistry campaigns, hidden models and fault handling [45]. Summit provides benchmark reactions and optimization strategies [46]. PyLabRobot supplies a hardware-agnostic control layer [47]. BixBench3 evaluates analysis on study-scale computational biology tasks [48]. A PCL contributor might reuse an evaluation concept from one work, a software interface from another, and local measurements for the oracle. These are three distinct provenance entries. The contribution record should retain all three, with version and license or authorization as appropriate.

An example contribution. For the three-point I–V exercise, a new contributor can register a device-specific points list, operating bounds, measurement model, target and tolerance. An additional synthetic thin-film adapter illustrates this extension in the repository. To replace the simple I–V response with evidence from a real device, a site could supply signed-current traces with an independently checked resistance and a contact-failure example. Split by device or specimen before selecting repeat sweeps. Reject an adapter that produces the right slope while accepting a documented unsafe configuration.

Reading the catalog. Each entry gives the capability the authors actually evaluate, the type of source and an original proposed adaptation. Conference and journal papers, preprints, software repositories, standards and project pages are identified separately. The bibliography gives direct source links; the BibTeX file retains full available author lists. Public program records are cataloged separately from scientific and software resources.

Data and code availability

The repository includes a searchable reference catalog, a machine-readable node map with individual award links, and a crosswalk from prior work to proposed tasks. The PDF, figures, bibliography and scripts are included with the code. Public readers can run the pilot without model credentials.

References

[1] US National Science Foundation NSF announces $400M investment in new national network of AI-programmable cloud laboratories in alignment with the U.S. government's Genesis Mission. *Program announcement*, 2026.

[2] Engineering Biology Research Consortium National Network of AI-Programmable Cloud Laboratories Summit (PCL Kickoff). *Summit program*, 2026.

[3] Northwestern Center for Synthetic Biology DREAM Cloud Lab for Protein Engineering. *Facility overview*, n.d.

[4] Yibo Wen and Han Liu Autonomous Science Bench: Evaluating AI Agents for Autonomous Scientific Discovery. *arXiv*, 2026.

[5] Yuyang Liu et al. BioProBench: A Corpus and Benchmark for Biological Protocol Reasoning in Autonomous Science. *International Conference on Machine Learning*, 2026.

[6] Odhran O'Donoghue et al. BioPlanner: Automatic Evaluation of LLMs on Protocol Planning in Biology. *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, 2023.

[7] Igor Ivanov BioLP-bench: Measuring understanding of biological lab protocols by large language models. *bioRxiv*, 2024.

[8] Jon M. Laurent et al. LAB-Bench: Measuring Capabilities of Language Models for Biology Research. *arXiv*, 2024.

[9] Jon M. Laurent et al. LABBench2: An Improved Benchmark for AI Systems Performing Biology Research. *arXiv*, 2026.

[10] Maokai Qin et al. SciHorizon-eLab: An Agentic Protocol-to-Task Compiler for Scalable Benchmarking of Scientific Embodied Agents. *arXiv*, 2026.

[11] Yankai Jiang et al. A Self-Evolving Agentic System for Automated Generation and Execution of Biological Protocols. *arXiv*, 2026.

[12] S. Hessam M. Mehr et al. A universal system for digitization and automatic execution of the chemical synthesis literature. *Science*, 2020.

[13] Anqi Zou et al. LabOSBench: Benchmarking Computer Use Agents for Scientific Instrument Control. *arXiv*, 2026.

[14] Indrajeet Mandal et al. Evaluating large language model agents for automation of atomic force microscopy. *Nature Communications*, 2025.

[15] Jiaqi Cai et al. Evaluating Instrument-Centered AI Agents for Table-Top Science Experiments. *arXiv*, 2026.

[16] Zhiqian Lan et al. AutoBio: A Simulation and Benchmark for Robotic Automation in Digital Biology Laboratory. *ICLR*, 2026.

[17] Rui Li et al. LabUtopia: High-Fidelity Simulation and Hierarchical Benchmark for Scientific Embodied Agents. *Advances in Neural Information Processing Systems*, 2025.

[18] Fei Lin et al. BEAKER: An Expert-Curated Benchmark for Embodied Brains in Self-Driving Chemical Laboratories. *arXiv*, 2026.

[19] Suozhi Huang et al. LabWorld: Learning Interactive 4D Worlds for Scientific Experimentation. *arXiv*, 2026.

[20] Fengshuo Bai et al. SafeLab: An Interactive High-Fidelity Benchmark for Embodied Safety in Scientific Robotics. *International Conference on Machine Learning*, 2026.

[21] Yujun Zhou et al. Benchmarking large language models on safety risks in scientific laboratories. *Nature Machine Intelligence*, 2026.

[22] Qianpu Sun et al. LABSHIELD: A Multimodal Benchmark for Safety-Critical Reasoning and Planning in Scientific Laboratories. *arXiv*, 2026.

[23] Yves Delley et al. Semantic Representation for Agentic Self-Play in Automated Laboratories. *arXiv*, 2026.

[24] jang1563 LabCraft-Eval. *Software repository*, 2026.

[25] Peter Jansen et al. DiscoveryWorld: A Virtual Environment for Developing and Evaluating Automated Scientific Discovery Agents. *Advances in Neural Information Processing Systems*, 2024.

[26] Ruoyao Wang et al. ScienceWorld: Is your Agent Smarter than a 5th Grader?. *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, 2022.

[27] Haonan Duan et al. Measuring Scientific Capabilities of Language Models with a Systems Biology Dry Lab. *Advances in Neural Information Processing Systems*, 2025.

[28] Chris Beeler et al. ChemGymRL: A customizable interactive framework for reinforcement learning for digital chemistry. *Digital Discovery*, 2024.

[29] Florian Häse et al. Olympus: a benchmarking framework for noisy optimization and experiment planning. *Machine Learning: Science and Technology*, 2021.

[30] Ben Reilly, Leah Damon and Daniel McKinnon LabBench: Can AI agents decide what experiment to run next?. *Gamow Labs*, 2026.

[31] Roman Bushuiev et al. MassSpecGym: A benchmark for the discovery and identification of molecules. *Advances in Neural Information Processing Systems*, 2024.

[32] Yuetong Wu and Maojun Sun AutoXRD: Autonomous LLM Agents and Comprehensive Evaluation for Powder Diffraction Analysis. *arXiv*, 2026.

[33] Kieran Didi et al. FLIP2: Expanding Protein Fitness Landscape Benchmarks for Real-World Machine Learning Applications. *International Conference on Machine Learning*, 2026.

[34] Daniil A. Boiko et al. Autonomous chemical research with large language models. *Nature*, 2023.

[35] Alexus A. Smith et al. Using a GPT-5-driven autonomous lab to optimize the cost and titer of cell-free protein synthesis. *bioRxiv*, 2026.

[36] SLE Team Science's Last Exam. *Science's Last Exam project*, 2026.

[37] Bioprotocols Working Group The Laboratory Open Protocol Language (LabOP). *Official project documentation*, n.d.

[38] SiLA Consortium SiLA 2 standard: core and mapping specifications. *Official specification*, n.d.

[39] Opentrons Python API Reference: Executing and Simulating. *Vendor documentation*, n.d.

[40] Anthropic Previewing the Model Hardware Standard. *Research preview*, 2026.

[41] Benjamin Burger et al. A mobile robotic chemist. *Nature*, 2020.

[42] Nathan J. Szymanski et al. An autonomous laboratory for the accelerated synthesis of inorganic materials. *Nature*, 2023.

[43] Junyu Chen et al. An AI-powered cloud biofoundry for autonomous biological research. *bioRxiv*, 2026.

[44] luhr BioFaudry laboratory digital twin assets. *Hugging Face dataset card*, 2026.

[45] Jiangjie Qiu et al. ChemWorld: Programmable Chemical Worlds for Controlled and Replayable Agent Experimentation. *arXiv*, 2026.

[46] Kobi C. Felton et al. Summit: Benchmarking Machine Learning Methods for Reaction Optimisation. *Chemistry–Methods*, 2021.

[47] Rick P. Wierenga et al. PyLabRobot: An open-source, hardware-agnostic interface for liquid-handling robots and accessories. *Device*, 2023.

[48] Zane Koch et al. BixBench3: Benchmarking AI agents on research-study-scale computational biology tasks. *arXiv*, 2026.

[49] Brandon J. Reizman et al. Suzuki–Miyaura cross-coupling optimization enabled by automated feedback. *Reaction Chemistry & Engineering*, 2016.