

From Experiment Execution to Verifiable Capability: An Assurance Architecture for Self-Driving Laboratories

Chao Wang*, Yanfeng Li, Yuchi Shen, Yuming Chen, Jianqing Li, Bin Liu

Engineering Research Center of Intelligent Theranostics Technology and Instruments, Ministry of Education, School of Biomedical Engineering and Informatics, Nanjing Medical University, Nanjing, 211166, China.

* Corresponding author: chaowang@njmu.edu.cn.

Abstract

Self-driving laboratories increasingly select experiments, orchestrate resources, execute protocols and update scientific models. Yet existing architectures often under-specify a critical runtime question: what evidence shows that the current physical system can satisfy a particular task under its operating conditions? Device availability, command completion and internal feedback do not establish that delivered volume, end-effector pose or sample temperature meets task-specific limits. Periodic calibration provides essential baseline evidence, but cannot support every operation through a single static pass label. We propose a three-layer architecture comprising task planning and scheduling, task-level capability assurance, and execution and physical operations. The assurance layer translates formal tasks into machine-readable capability requirements, represents measurement evidence through versioned, uncertainty-aware records with explicit applicability and validity, and uses these records to admit, constrain, reject, reassign or trigger verification of tasks. Maintenance, collisions, environmental excursions and version changes dynamically invalidate affected records. Parameter correction is treated as an optional bounded recovery process requiring authorization, readback, independent re-verification and commit or rollback. Examples in liquid handling, robotic positioning and temperature control illustrate the framework. Trustworthy self-driving laboratories must therefore maintain machine-actionable evidence that they are currently qualified to perform each task.

Keywords: autonomous laboratory; self-driving laboratory; task-level capability assurance; capability evidence; task admission; metrological traceability; measurement uncertainty

1. Introduction

Here, we use the term self-driving laboratories to refer to autonomous laboratory systems that integrate experiment selection, automated execution, measurement and model updating in a closed loop [1-4]. Mobile robotic chemists, automated synthesis platforms, and materials laboratories have shown that physical experimentation can be coupled to long-running optimization loops [5-8]. Similar ideas are entering protein engineering and other biological search spaces [9]. Orchestration software, modular workflow frameworks, knowledge graphs, and language-model interfaces are lowering the barrier to coordinating heterogeneous instruments [10-14]. In the life sciences, biofoundries and automated design-build-test-learn (DBTL) pipelines have likewise demonstrated that complex biological workflows can be standardized, scaled, and programmed [15-17].

National policy initiatives indicate that AI-enabled experimentation is moving from isolated demonstrations toward coordinated infrastructure and sector-wide deployment. The U.S. Genesis Mission calls for an integrated platform connecting scientific datasets, domain-specific foundation models, AI agents, high-performance computing, and experimental and production tools, including robotic laboratories for AI-directed experimentation [18]. In parallel, China’s 15th Five-Year Plan for the Pharmaceutical Industry promotes high-quality pharmaceutical datasets, trustworthy health-data spaces, pharmaceutical-domain large models and intelligent agents, AI-enabled drug R&D and quality management, high-end sensing and online analysis, and validation services for digital transformation [19]. These initiatives establish strategic demand for more autonomous scientific and biomedical workflows. However, platform integration, intelligent agents, and digital quality systems do not by themselves establish whether a particular physical resource is currently capable of executing a specific task within defined uncertainty, applicability, identity, version, and validity constraints.

Many accounts of laboratory autonomy focus on two hard questions. The first concerns what should be done next, covering hypothesis generation, experiment selection, active learning, and optimization. The second addresses how the selected experiment should be executed, including protocol decomposition, scheduling, resource allocation, device control, and data acquisition.

Both are essential. Yet taken together, they leave a third question insufficiently addressed: what evidence confirms that the physical system is adequately calibrated and qualified to perform this specific task at the required standard?

This question becomes consequential when laboratories move from supervised automation toward multi-device, long-duration, minimally attended operation. A liquid handler may report normal piston travel or pressure traces while its realized transfer volume has shifted because of tip lots, liquid properties, seal wear, or aspiration conditions. A robot may report valid joint positions while the tool center point has moved relative to a plate, centrifuge, or incubator interface after a collision or tool change. A temperature controller may hold its internal probe at the set point while a representative sample experiences a different temperature because of load, container geometry, airflow, or spatial gradients. The controller confirms an internal state or an issued action; the experiment depends on an external physical result. These are related, but they are not equivalent.

The distinction matters especially in biological workflows. Small physical deviations can propagate through plates, incubation periods, and downstream analyses. Some operations irreversibly alter scarce samples before a failure becomes visible. When capability loss is detected only after an assay is complete, the problem is not merely how to restore the device; the laboratory must also determine which samples and results produced since the last credible verification remain trustworthy.

Here, we propose a conceptual and testable three-layer architecture for trustworthy autonomous laboratories. The architecture separates task planning and scheduling, task-level capability assurance, and execution and physical operations. Its defining feature is a capability-assurance layer between planning and execution that connects task-specific requirements to independent physical evidence and runtime admission decisions. This layer does not replace scientific planners, device controllers, calibration procedures, or quality systems. The proposal is therefore an architectural perspective, not a claim that one measurement technology or one self-calibration algorithm can make an entire laboratory trustworthy.

This perspective makes three contributions. First, it defines task-level capability evidence as a

machine-actionable object rather than a static device label. Second, it specifies uncertainty-aware admission, event-triggered invalidation, and bounded recovery logic. Third, it proposes a testable benchmark structure for evaluating admission correctness, evidence coverage, failure detection, and recovery integrity. The proposal is architectural and testable, but not yet a universally validated implementation.

2. The missing object is task-level evidence

2.1 Command completion is not task capability

An execution controller typically verifies four core states: command acceptance, actuator operation within specified limits, internal sensor setpoint tracking, and workflow attainment of the intended state. A task-level capability decision asks something different: under the present combination of device, tool, consumable, location, load, environment, and parameter version, did the realized physical operation meet the target and tolerance required by this task?

This is the difference between observing that a process ran and establishing that the result is fit for its intended use. Rich internal telemetry can detect many faults, but it can share sensors, models, coordinate frames, calibration parameters, and common failure modes with the execution mechanism. Additional internal alarms improve diagnosis; they do not automatically create independent evidence about the final measurand. Equipment health, workflow state, and task capability should therefore be represented separately. A device can be online and responsive while lacking evidence that covers the current task, and a valid measurement can show that an otherwise operational device is not qualified.

2.2 Periodic calibration is necessary but not sufficient for runtime admission

Calibration, intermediate checks, and quality-control programs provide essential information about equipment performance. Measurement science supplies concepts of metrological traceability, uncertainty, and conformity assessment [20-22]. Automated liquid handling is supported by standards for volumetric performance and reference measurement procedures [23,24], while laboratory competence is framed more broadly by quality-system requirements such as ISO/IEC 17025 [25]. These frameworks address essential parts of the problem, including calibration, measurement competence, quality control, and traceability. An under-specified architectural gap remains around a direct, computable connection between their outputs and autonomous scheduling decisions.

First, an unattended platform cannot comprehensively test every physical operation immediately before use. Such “test every time” logic would consume time, reagents, reference materials, and device life, eroding the throughput and economic value of autonomy. The practical question is when existing evidence may be reused, which changes invalidate it, and what smallest task-relevant verification is needed when coverage is insufficient.

Second, different tasks impose different requirements on the same device. A routine wash dispense, a low-volume quantitative addition, and an irreversible delivery of a critical reagent may use the same channel but differ markedly in tolerance, risk, and required evidence strength. Periodic calibration establishes baseline general capability under stated conditions; it does not automatically create task-specific admission criteria for every level of difficulty and consequence. The same performance record may be sufficient for a low-risk task and inadequate for a narrow-tolerance task.

Third, conventional certificates or pass/fail labels rarely express all conditions needed for automated reuse: channel, tool, liquid class, consumable lot, work region, payload, environmental range, uncertainty, firmware and parameter versions, validity interval, and invalidating events. Digital calibration certificates are making metrological information structured and machine-readable [26,27]. Nevertheless, digitizing a certificate is not the same as making a task admissible. A runtime system must still connect the measurement result to a task requirement, an applicability domain, and an explicit decision rule.

Maintenance, collision, device relocation, tool replacement, software updates, environmental excursions, and accumulated operation can all break the assumptions under which earlier evidence was obtained. Calendar-based recalibration can address some gradual drift, but it does not express immediate event-triggered loss of validity. Judgments previously made tacitly by experienced staff must become explicit, auditable state transitions in an unattended system.

2.3 False admission and the non-negligible risk of false rejection

A device deviation becomes a scientific failure when an under-capable device is admitted to a formal task. We use “false admission” to mean authorizing a task when the true capability does not meet its requirement; in conformity-assessment terminology, this is analogous to false acceptance. In long cell-based workflows, screening campaigns, or clinical-sample processes, the failure may remain hidden for hours or days and affect an entire batch of samples and conclusions.

The opposite error, false rejection, also matters. An overly conservative policy can stop equipment that is actually capable, reduce throughput, consume scarce material, and create excessive human escalation. Capability assurance is therefore not a demand for impossible per-operation proof. It is a transparent allocation of evidence under competing risks: false admission, false rejection, and verification cost. Runtime-assurance research in safety-critical autonomous systems similarly separates an advanced controller from an independent mechanism that can constrain or withdraw authority [28]. Laboratory assurance extends this principle by making physical measurement, task tolerance, and experimental consequence part of the runtime decision.

3. A three-layer architecture for trustworthy autonomy

We propose three logical layers (Fig. 1). A layer denotes responsibility and authority, not necessarily a separate computer or physical enclosure. Multiple functions may share infrastructure, but their interfaces and decision rights must remain explicit.

3.1 Layer 1: task planning and scheduling

The task planning and scheduling layer answers what to do and how to organize it. It receives a scientific objective, selects or generates an experimental plan, decomposes that plan into formal tasks, and allocates instruments, samples, consumables, and time windows. For each critical physical task, it should also declare the minimum capability requirement: target or target interval, allowable deviation, repeatability or stability requirements, operating conditions, risk class, and scientifically acceptable alternatives.

For example, “dispense 10 μ L into each well of a 96-well plate” is not yet a complete admission request. The planning layer may also need to identify liquid class, volume tolerance, destination geometry, tip type, timing constraints, and whether the step is irreversible. The planning layer declares what is required; it does not certify that a chosen device satisfies it.

3.2 Layer 2: task-level capability assurance

The middle layer is the core of the architecture. It (i) parses formal tasks into capability requirements; (ii) queries and maintains capability evidence; (iii) checks applicability domain, identity, version, and validity; (iv) applies uncertainty-aware admission rules; (v) generates verification tasks when evidence is missing; and (vi) withdraws admission and coordinates safe disposition when assumptions change.

This layer has independent refusal authority. Even if the planner has selected a device and the executor reports that it is online, the assurance layer may deny release because evidence is absent, expired, or out of scope. Its response should be more informative than a binary pass/fail. Possible outcomes include admit, admit with constraints, reassign to another resource, narrow the task if scientifically acceptable, run additional verification, isolate, or request human review.

The layer may use embedded sensing, external metrology modules, shared verification stations, and process-quality data. Evidence independence should be assessed against common failure modes rather than physical location alone. Useful dimensions include sensing independence, reference independence, acquisition independence, and decision independence. An internal pressure trace may be valuable supporting evidence, but if it shares the same failure chain as the liquid-handling mechanism, it should not alone establish delivered volume.

Here, independence is used as a failure-mode-relative engineering property rather than as a claim of absolute statistical independence or physical separation. A measurement path is sufficiently independent for a given task when it does not share the critical sensor, reference, acquisition chain, coordinate or parameter state, or decision logic whose common failure could produce the same erroneous admission. Embedded, external, or shared measurement arrangements may therefore qualify, provided that the relevant common-cause failure modes and the rationale for independence are explicitly recorded.

3.3 Layer 3: execution and physical operations

The execution and physical operations layer contains liquid handlers, robotic arms, incubators, centrifuges, thermal modules, and analytical instruments. It performs admitted formal tasks as well as verification tasks requested by the middle layer. It returns internal feedback, events, and physical outcomes. Independent metrology in or alongside this layer observes task-relevant quantities such as transferred volume, end-effector pose, or representative sample temperature.

The execution layer cannot promote an internal “success” flag to a task-level qualification and cannot silently bypass a withdrawn admission. Device control remains local to the executor, while permission to use that device for a particular task is constrained by the assurance layer.

4. From measurements to machine-actionable capability records

4.1 Task capability requirements

An executable step can be represented as a task capability requirement

$$R = (o, d, \theta, L, \rho, \Omega_R, V_R, \Delta t)$$

where o identifies a candidate device or object, d is the capability dimension, θ is the target or target interval, L is the task limit, ρ is a risk class, Ω_R is the required operating domain, V_R identifies required tool, firmware, model, and parameter versions, and Δt is the relevant time window.

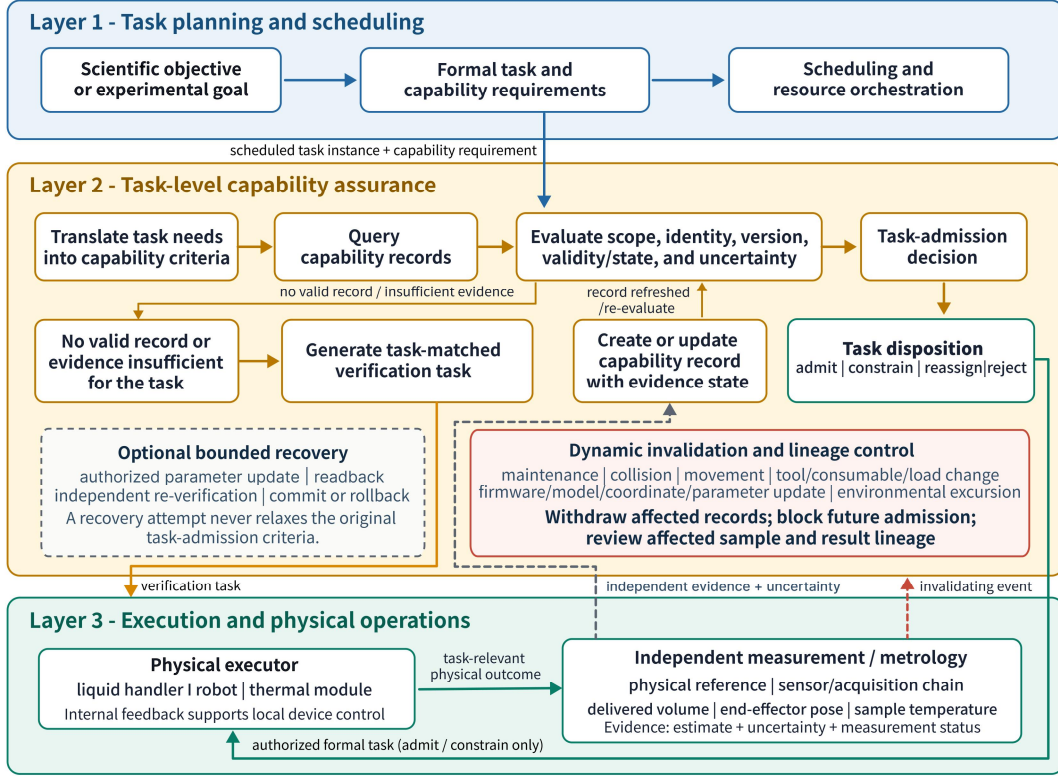


Figure 1. Three-layer architecture for trustworthy autonomous laboratories. The planning layer translates scientific objectives into formal tasks and capability requirements. The assurance layer matches these requirements to versioned, domain-bounded, and uncertainty-aware evidence and determines admission, constrained use, reassignment, rejection, or verification. The execution layer performs formal and verification tasks and returns internal feedback and independent physical measurements. Invalidation withdraws affected records and triggers lineage review, while bounded recovery commits a new state only after independent re-verification against the original criteria.

The requirement should be derived from scientific need, method validation, and risk analysis rather than from the current performance of a convenient device. Otherwise, a system could effectively relax a task to admit the equipment that happens to be available. The same pipetting channel may therefore receive different requirements for a 10 μL critical-reagent addition and a 200 μL wash transfer.

4.2 Evidence and capability records

Raw observations, reference-object information, measurement models, and uncertainty budgets form capability evidence. After measurement-quality checks and a stated decision rule, the system represents reusable evidence as a capability record

$$C = (o, d, \hat{y}, u, \Omega_C, V_C, \tau, I, s_q, s_d, s_m, P)$$

where $\hat{y}$ is an estimated performance value, u is measurement uncertainty, Ω_C is the verified applicability domain, V_C binds the record to relevant versions, τ describes time- or count-based validity, I is a set of invalidating events, s_q , s_d , and s_m denote qualification, diagnostic, and measurement-quality states, and P records provenance and audit information.

For a machine-actionable record, P should contain immutable identifiers and versions for the reference, measurement model, uncertainty budget, independence assessment, and conformity or decision rule, together with the evidence lineage.

The three states should not be collapsed. Qualification may be unverified, qualified, conditionally qualified, expired, or withdrawn. Diagnosis may indicate conformity, an identifiable and

potentially correctable systematic deviation, a non-correctable fault, or an unknown cause. Measurement quality may be valid, low-confidence, or invalid. A measurement can be valid yet show non-conformity; a deviation can appear correctable while the measurement supporting that diagnosis is invalid.

Direct admission requires a qualification state that permits use, no unresolved diagnostic blocking condition, and valid measurement quality. A conditionally qualified state permits only explicitly bounded use. Withdrawn, expired, out-of-domain, non-correctable, or measurement-invalid states withhold admission. A correctable deviation may enter bounded recovery but cannot support direct admission. An invalid measurement withholds evidence but does not by itself establish capability nonconformity; nonconformity requires valid and traceable evidence showing that the applicable requirement is not satisfied.

The minimum trustworthy unit is therefore not “device qualified”, but a device-capability-condition-version-evidence combination. A record should cover only conditions directly verified or supported by a pre-validated interpolation model. Evidence from one channel, location, liquid class, payload, or temperature regime should not be extrapolated to an entire instrument without justification.

4.3 Uncertainty-aware task admission

Admission is a match between the task requirement R and the capability record C , rather than a generic qualification label assigned to a device. A task can be admitted only when the object identity and required tool, firmware, model, and parameter versions agree; the required operating domain Ω_R is covered by the verified domain Ω_C ; the capability record remains valid during the relevant time window and event state; the measurement quality satisfies the applicable risk policy; and the uncertainty-aware capability boundary lies within the formal task limit.

For a scalar capability dimension, let $\hat{y}$ denote the estimated value, θ the target value, u the combined standard uncertainty, k a coverage factor selected according to the required confidence and risk policy, and L the allowable deviation specified by the task. A conservative guard-band quantity can then be written as

$$M = |\hat{y} - \theta| + ku$$

If $M \leq L$, and any additional repeatability, stability, uniformity, or model-residual criteria are satisfied, the capability record may support task admission. This expression is an illustrative decision rule rather than a universal conformity-assessment rule. The decision quantity and coverage factor should be selected according to the measurement model, the uncertainty budget, and the relative consequences of incorrect acceptance and rejection [22]. For an interval target or asymmetric limits, the corresponding distance-to-interval or one-sided decision rule should be used instead of the absolute-deviation form.

Many laboratory capabilities are multidimensional. Let j index the relevant capability dimensions, and let $\hat{y}_j$, θ_j , u_j , and L_j denote the estimated value, target value, combined standard uncertainty, and allowable deviation for dimension j , respectively. A componentwise worst-case rule may be expressed as

$$M_{multi} = \max_j \frac{|\hat{y}_j - \theta_j| + k_j u_j}{L_j}$$

Under this rule, admission requires

$$M_{multi} \leq 1$$

together with any additional requirements for repeatability, stability, covariance, uniformity, or

operating-domain coverage. The max-normalized expression is appropriate when the task limits are specified separately for each dimension and the decision policy is governed by the worst normalized margin. When capability dimensions are correlated or a joint confidence region is required, a covariance-aware multivariate decision rule should be used instead; independence between dimensions should not be assumed.

Importantly, L and L_j are the formal task limits derived from scientific requirements, method validation, and risk analysis. If an automatic-correction eligibility threshold is defined, it is a separate, safety-bounded threshold that governs whether a recovery attempt may be initiated. It does not relax the original admission requirement. Any correction should be subject to authorization and pre-write checks, readback, scope control, independent re-verification, and rollback where necessary. After correction, the original scalar or multidimensional admission rule must be satisfied again before the task is released.

5. Admission as an evidence lifecycle

The main workflow is: task requirement $\rightarrow$ capability-record query $\rightarrow$ scope, version, state, and uncertainty matching $\rightarrow$ admission $\rightarrow$ physical execution $\rightarrow$ evidence update. Its practical value depends on explicit branches for missing evidence, failed verification, and changing conditions (Fig. 2).

5.1 Three basic admission paths

First, when a valid capability record covers the current task, the system can admit it directly. Comprehensive re-testing before every operation is unnecessary; the reusable record is what makes evidence-based autonomy efficient.

Second, when no valid record exists but verification is feasible, the assurance layer generates a task-matched verification. If this first verification passes directly, the system creates a capability record and admits the formal task. Verification does not imply that a parameter correction must occur.

Third, when verification fails, evidence remains insufficient, or measurement quality is invalid, the formal task must not proceed by default. Invalid measurement withholds the evidence but does not by itself establish capability nonconformity. The system should first inspect the reference, acquisition chain, and measurement model, and may repeat or replace the measurement. Only valid, traceable evidence showing failure of the applicable requirement supports a capability-nonconformity decision. Depending on task substitutability, sample value, risk, and time constraints, the system can reject or reassign the task, request targeted verification, isolate the resource, or escalate to a person.

Verification can occur at multiple time scales: immediate checks before high-risk actions, pre-run and post-run bracketed verification, checkpoints within long workflows, statistical sampling, periodic verification, and event-triggered verification. Two successful checks bracketing a run may support an interval-level capability claim only under an explicit assumption that no unobserved invalidating event occurred between the checkpoints and that the drift model and checkpoint spacing satisfy a prespecified bound. They do not prove that every intermediate action was correct.

5.2 Dynamic invalidation and selective withdrawal

Capability records must be revocable. Elapsed time or operating counts, changes in tools,

consumables, references, loads, or work regions, firmware or model updates, coordinate-transform or parameter changes, device movement, collision, emergency stops, maintenance, component replacement, and environmental excursions can all invalidate a record. The assurance layer should subscribe to these events and withdraw affected admissions rather than wait for the next scheduled calibration.

Invalidation should be selective where evidence supports selectivity. Replacing the seal in one liquid-handling channel need not invalidate every capability of the workstation; replacing a robotic end effector need not affect unrelated tools. Fine-grained applicability domains reduce unnecessary downtime, provided that identities, versions, and dependencies are reliably recorded.

5.3 Post-run failure and the affected interval

A failed checkpoint or post-run verification should first trigger validation of the reference and measurement chain. If the evidence is valid and indicates capability loss, future tasks remain blocked until capability is re-established. Samples, plates, operations, and results since the last credible verification point, or from a more conservative boundary when the onset of capability loss is not observable, form an affected interval. Risk-based dispositions include targeted re-measurement, quarantine, repeat experimentation, release with justification, or human review.

This traceability is important in path-dependent biological workflows. A capability event is not only a maintenance issue; it defines a computable boundary on data trustworthiness when linked to sample lineage and the experimental event log.

6. Automated correction is an optional, bounded recovery branch

Equating capability assurance with automatic calibration would obscure the central contribution. Most admissions should reuse valid records, and an initial verification may pass without intervention. Automatic recovery is appropriate only when measurement quality is acceptable, the deviation is stable and identifiable, an authorized parameter has a validated relationship to that deviation, the affected scope is understood, and independent re-verification and rollback are available.

Pre-write gates should confirm device and parameter identity, measurement uncertainty and repeatability, deviation direction and identifiability, parameter authorization, single-step and cumulative limits, recoverability of the original parameter version, safety interlocks, and termination conditions. If any gate fails, the system should stop automatic correction and move to rejection, reassignment, isolation, or human handling.

When the gates pass, recovery should be transactional:

save original → trial write → readback → independent re-verification → commit if the
original admission criteria pass; otherwise rollback

Readback occurs after trial writing and is not a pre-write gate. It verifies only that the candidate parameter value was written successfully; it does not demonstrate that physical capability has been restored. The candidate state remains provisional until independent re-verification passes against the original admission criteria. The system should prohibit an unbounded "adjust until it passes" loop, which can hide random variation, hardware degradation, reference failure, or model inadequacy. If rollback fails, the affected resource must remain isolated and the task must be reassigned or escalated for human handling.

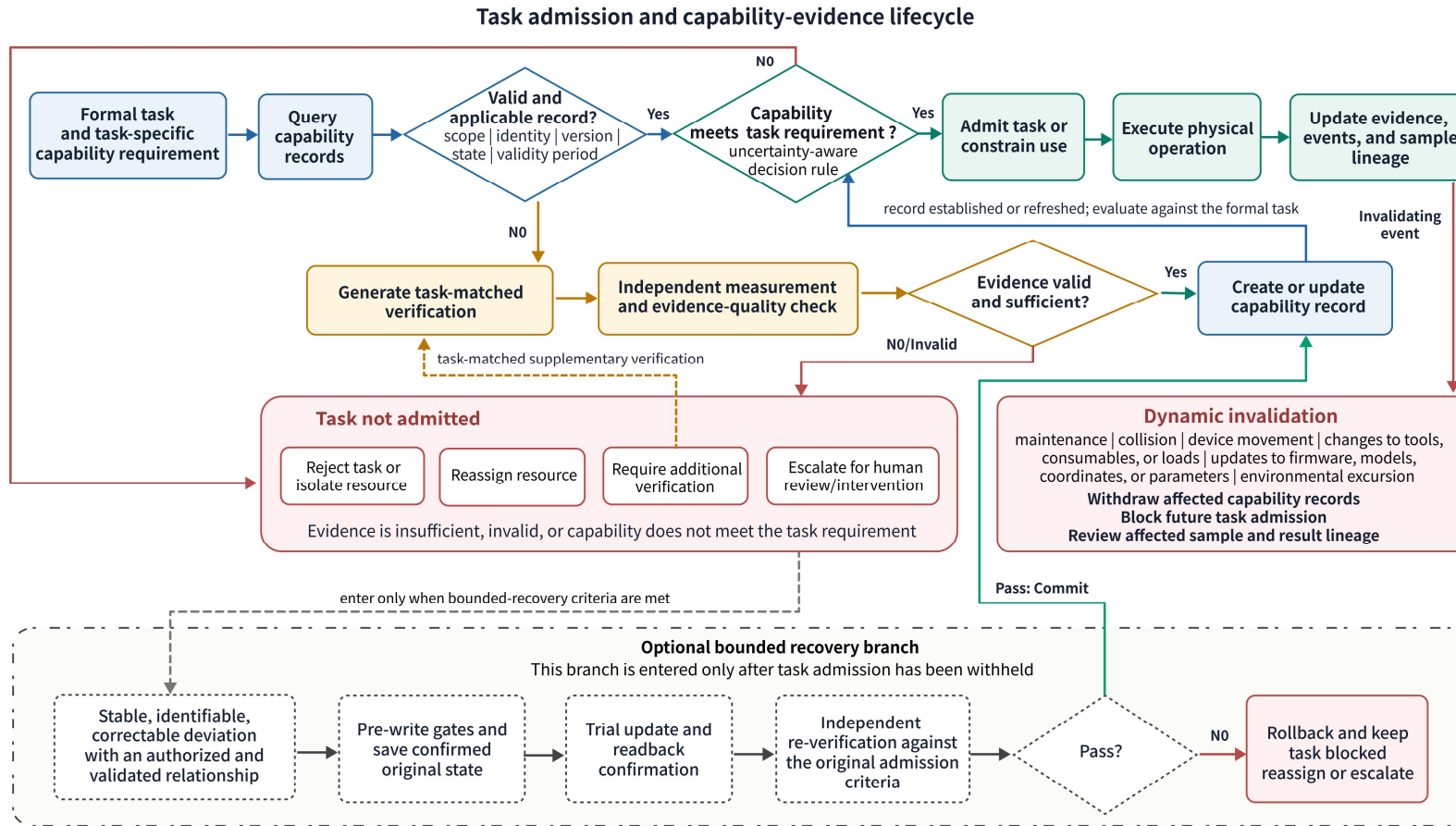


Figure 2. Task admission and capability-evidence lifecycle. Task requirements are matched to valid capability records; missing or insufficient evidence triggers task-matched verification. Evidence updates the record and supports reassessment, whereas invalid evidence or state changes withhold or withdraw admission and trigger appropriate disposition and lineage review. Correctable deviations may enter bounded recovery, with admission restored only after independent re-verification against the original criteria.

This branching logic clarifies what “unattended” should mean. It does not mean that a human is never requested. It means that routine decisions, authority boundaries, and escalation conditions are formalized in advance, allowing the system to continue, refuse, reassign, verify, recover, or stop safely without continuous supervision. When cause is ambiguous or consequence is severe, preserving state and requesting human intervention can be the correct autonomous action.

7. Cross-device illustrations

The three-layer architecture supplies a common decision logic but does not impose one measurement technology. Liquid handling, robotic positioning, and temperature control involve different measurands, references, and applicability domains (Table 1).

Table 1. Illustrative task-level capability assurance for three automation domains.

Domain	Task capability	Evidence and applicability domain	Example recovery boundary
Liquid handling	Target volume, allowable bias, precision, liquid class, and task risk	Gravimetric or photometric volume estimate, uncertainty, and repeatability, bound to channel, tip type/lot, liquid class, volume range, vessel, deck region, environment, and parameter version	Correct only stable, identifiable systematic bias linked to an authorized liquid parameter; do not compensate for leakage, clogging, excessive random error, or invalid measurement
Robotic positioning	Position, orientation, repeatability, payload, and placement tolerance in an external laboratory frame	External vision, fiducial, or dimensional measurement, bound to end effector, installation, payload, region, pose, approach direction, speed, and coordinate version	Apply authorized local or coordinate correction only in a re-verified pose domain; do not extrapolate a local result to the full workspace
Temperature control	Representative sample temperature, stability, spatial uniformity, ramp time, and hold profile	Independent temperature measurement in a representative load, bound to vessel, load distribution, region, target program, environment, controller, and compensation version	Correct a uniform, stable, repeatable offset only; do not conceal gradients, persistent oscillation, or inadequate heating/cooling capacity with one offset

7.1 Liquid handling as a candidate proof-of-concept domain

Automated liquid handling is a strong first proof-of-concept domain. Internal piston and pressure feedback are clearly related to, but not identical with, delivered volume. Gravimetric and photometric methods, international standards, and extensive engineering practice provide a foundation for independent evidence [23,24,29-30].

A proof of concept should prospectively compare a static device-level pass label with the proposed task-level admission policy while varying target volume, liquid class, tip lot, channel, environment, operating count, task tolerance, and parameter version. An independent reference measurement should define the true capability state. Controlled fault injection, for example, seal degradation, tip-lot change, parameter-version mismatch, or reference failure, can test selective invalidation, channel reassignment, and targeted verification. Operation-, run-, plate-, and sample-level units

should be distinguished to avoid pseudoreplication. False-admission and false-rejection rates, evidence coverage, detection latency, affected-interval length, recovery integrity, rollback success, and assurance overhead should be reported with explicit denominators and confidence intervals. The scientific question is whether measurement evidence changes runtime scheduling correctly, not merely whether volume can be measured again.

7.2 Robotic positioning: capability is regional and task-dependent

A robot controller regulates internal joint space, while laboratory tasks occur in an external workspace. Passing at one reference pose does not imply that all payloads, orientations, approach directions, and regions meet the same requirement. Robotic capability records should therefore bind the tool, coordinate version, payload, and verified pose set.

When a stable local deviation appears, the assurance layer can generate diagnostic poses to test whether the error has an identifiable relationship to an authorized local or coordinate parameter. If candidate causes remain coupled, repeatability is poor, or hardware damage is plausible, automatic correction should stop. A perspective need not prescribe a universal parameter-identification algorithm; it should specify the evidential and authority conditions under which correction is allowed.

7.3 Temperature control: controller display is not sample history

For incubators, thermal modules, and temperature-controlled storage, the scientifically relevant quantity is the temperature experienced by the sample or a representative load, not merely the controller probe. Independent temperature measurements can distinguish a uniform offset from spatial non-uniformity, dynamic oscillation, and delayed equilibration.

An overall offset is a candidate for bounded correction only when representative locations show the same direction of deviation, between-location differences lie within a predeclared uniformity limit, and stability and ramp-time requirements pass. Terms such as “similar values” or “significant differences” must be defined by task-level thresholds or statistical decision rules rather than interpreted ad hoc. Different abnormal patterns require different dispositions, reinforcing the need to separate qualification, diagnosis, and measurement quality.

8. A research agenda

8.1 Capability ontologies and machine-actionable records

Existing laboratory interfaces primarily describe commands, resources, and data formats. Autonomous laboratories also need a shared way to express what capability, under which conditions, supported by what evidence, can be trusted for which task. A minimum record should contain capability dimension, estimate, uncertainty, applicability domain, identity, versions, validity conditions, invalidating events, provenance, and decision state. FAIR principles provide a foundation for reusable scientific data [31]; digital calibration certificates and laboratory knowledge graphs offer complementary models for structured metrology and experimental semantics [13,26,27].

The challenge is not to maximize the number of fields. It is to define a cross-device core with domain extensions for liquid handling, robotics, temperature control, and analytical instruments. Regulated settings will also require access control, signatures, and tamper-evident audit history.

8.2 Compact verification with sufficient coverage

If every new task demands a full recalibration, the architecture will not be useful. Research is needed on selecting the smallest set of reference volumes, poses, loads, or temperature locations that provides adequate coverage of a requested operating domain. This is an evidence-design problem balancing verification cost, false-admission risk, and the permitted scope of interpolation. Protocol semantics can drive verification. A high-risk low-volume dispense can trigger a check with the same channel, tip type, volume range, deck region, and representative liquid-property class. A robot using a newly installed tool may require verification only in the work regions dependent on that tool. Verification designs should be declared before observing the result to avoid outcome-dependent criteria.

8.3 Risk-adaptive verification scheduling

Fixed schedules are easy to manage but can over-test low-risk tasks and under-protect high-risk tasks after a sudden event. Verification frequency could depend on task consequence, equipment history, drift rate, operating count, environmental change, sample value, and the cost of retrospective failure. Statistical process control, change-point detection, reliability models, and Bayesian updating may help choose verification times, provided that their assumptions and uncertainty are exposed to the admission policy.

8.4 Fault-injection benchmarks

Autonomous laboratories are commonly evaluated by discovery rate, throughput, or optimization efficiency [4,8]. Task-level capability assurance requires additional metrics (Table 2). The ADePT framework has begun to formalize evaluation of autonomous laboratory robotics [32]; assurance-specific benchmarks should extend this effort to evidence quality, admission correctness, invalidation, and recovery.

Shared testbeds should inject both obvious and ambiguous faults: pipette-seal degradation, tip-lot shifts, camera or fixture movement, tool reinstallation, payload changes, temperature gradients, reference-material errors, network interruption, and parameter-version mismatch. The objective is not to prove that an autonomous system never fails, but to measure whether it contracts authority correctly when evidence becomes inadequate.

For each benchmark metric, the unit of analysis and denominator should be declared in advance, and uncertainty or confidence intervals should be reported. A reference-defined capability state is required to estimate false-admission and false-rejection rates objectively.

8.5 Governance, quality systems, and human responsibility

Task-level capability assurance does not eliminate human responsibility. Domain experts and quality personnel still define task limits, risk classes, authorized parameters, decision rules, invalidating events, and escalation policies. The system is responsible for applying those rules consistently and explaining why a task was admitted or denied and which evidence supported the decision.

In in vitro diagnostics, drug development, and cell-related applications, capability records could connect to method validation, deviation management, sample lineage, and audit trails. Deployment in regulated environments will require collaboration among metrologists, automation engineers, domain scientists, quality professionals, equipment manufacturers, and regulators. Databases of autonomous laboratory systems and reconfigurable workflow platforms provide

useful starting points for standardizing equipment and workflow descriptions [14,33]; task capability and evidence state remain missing complementary objects.

These policy directions also expose an assurance gap. Provenance tracking, online analysis, computerized-system validation, and quality-management infrastructure can organize and govern evidence, but they do not themselves define when that evidence is sufficient to release a specific physical task. The capability records and admission logic proposed here provide a candidate interface between institutional quality requirements and runtime autonomous control [18,19].

The proposed architecture does not by itself establish analytical validity, clinical validity, regulatory compliance, or authorization for clinical use.

Table 2. Candidate metrics for task-level capability assurance.

Metric	Interpretation
False-admission rate	Probability of permitting a task when true capability does not meet its requirement
False-rejection rate	Probability of unnecessarily denying a task when true capability is sufficient
Evidence coverage	Fraction of executed task conditions fully covered by currently valid records
Detection latency	Time or operation count from capability loss to admission withdrawal
Affected-interval size	Number of samples or operations requiring review after failed post-run verification
Recovery integrity	Whether target capability is restored without degrading other verified capabilities or safety margins
Rollback success	Fraction of failed candidate updates that reliably return to the confirmed original state
Assurance overhead	Time, reagent, wear, and opportunity cost introduced by verification
Audit completeness	Ability to reconstruct each decision from task, evidence, versions, and state transitions

9. Rethinking autonomy in the laboratory

“Unattended” describes whether a person is continuously present. “Autonomous” describes whether a system can make and execute decisions within authorized boundaries. A platform may run for long periods without people while repeatedly relying on stale parameters or expired evidence. Such a system is operationally automated but not necessarily scientifically trustworthy.

We distinguish three useful forms of autonomy for analyzing trustworthy laboratory operation: epistemic autonomy to decide what to learn or test, operational autonomy to organize and execute experiments, and assurance autonomy to determine whether the current physical system is qualified and to contract authority when evidence expires. The three-layer architecture does not

require the same algorithm for all three, but it requires explicit interfaces among them. Capability records may also improve scientific inference. When experimental results are linked to verified operating domains, versions, and evidence states, learning algorithms can better distinguish biological variability from equipment-state changes. Conversely, an advanced active-learning algorithm coupled to an executor whose capability has drifted can optimize the wrong physical process with increasing confidence. Capability assurance is therefore not merely a safety add-on; it is infrastructure for reducing hidden experimental confounding and improving reproducibility.

10. Boundary conditions and limitations

Independent measurement is not automatically correct. References can degrade, sensors and acquisition chains can drift or fail, and measurement models can be applied outside their valid domains. The assurance layer must therefore maintain its own traceability, uncertainty, version, calibration, and measurement-quality states. High-risk settings may require a partially independent second evidence pathway or external proficiency testing.

Not every biological capability can be verified before a task. Cell state, long-term culture outcomes, and some functional endpoints become observable only after hours or days. The architecture must support graded evidence: pre-admission for measurable physical capabilities, process monitoring and checkpoints for evolving biological states, and post-run evaluation for delayed endpoints. It provides bounded and explainable confidence, not absolute proof of every operation.

There is also a granularity trade-off. Highly specific applicability domains increase data and maintenance burden; overly broad domains invite unjustified extrapolation. Empirical studies must identify which conditions materially affect each capability and which can be safely grouped. The benefit of assurance must be compared with its operational overhead rather than assumed.

Finally, this perspective proposes a testable architecture, not a universally validated implementation. Its value should be quantified through prototypes and fault-injection experiments. A practical route is to begin with a measurable, high-consequence capability supported by existing standards (e.g., task-level volumetric admission in automated liquid handling) and then extend to robotic positioning, temperature control, and more complex biological processes.

11. Conclusion

Autonomous laboratories are rapidly gaining the ability to choose experiments, call instruments, and learn in closed loop. Their next bottleneck is not only a smarter planner or a more flexible robot; it is runtime evidence. Completing an action does not prove that a device satisfied the current task, and passing periodic calibration does not mean that one conclusion covers every task, condition, and time point.

The proposed three-layer architecture inserts task-level capability assurance between planning and physical execution. It turns tasks into capability requirements, organizes independent measurements into records with applicability domains, uncertainty, versions, and states, and uses them to admit, reject, reassign, verify, invalidate, or recover. Its purpose is not to make a laboratory calibrate itself continuously. It is to make the laboratory know when it may act, when evidence is insufficient, and when authority must be withdrawn.

The next generation of trustworthy autonomous laboratories should answer two questions together: “What experiment should be performed next?” and “What evidence shows that the physical

system is qualified to perform it now?” Making the second question machine-actionable is a prerequisite for autonomous experimentation that is verifiable, auditable, and scientifically worthy of trust.

Author contributions

Chao Wang conceived the core viewpoint, developed the overall argumentative framework, supervised the project, and critically revised the manuscript. Yanfeng Li drafted the original manuscript. Yuchi Shen conducted the literature review and organized the supporting evidence. Yuming Chen developed the analytical logic and prepared all conceptual figures and schematic diagrams. Jianqing Li validated the core arguments, provided intellectual feedback, and reviewed the manuscript. Bin Liu contributed to the revision of the manuscript and coordinated the project administration. All authors read and approved the final version of the manuscript.

Competing interests

Chao Wang is listed as an inventor on intellectual property related to capability assurance in automated laboratories. All other authors declare no competing financial or non-financial interests. This intellectual property does not affect the viewpoints presented in this perspective article.

Data and code availability

No new research datasets or analysis software were generated for this perspective. Editable figure sources and manuscript files accompany the preprint package.

References

1. King RD, Whelan KE, Jones FM, Reiser PGK, Bryant CH, Muggleton SH, et al. Functional genomic hypothesis generation and experimentation by a robot scientist. *Nature*. 2004;427(6971):247-252. <https://doi.org/10.1038/nature02236>
2. King RD, Rowland J, Oliver SG, Young M, Aubrey W, Byrne E, et al. The automation of science. *Science*. 2009;324(5923):85-89. <https://doi.org/10.1126/science.1165620>
3. Häse F, Roch LM, Aspuru-Guzik A. Next-generation experimentation with self-driving laboratories. *Trends in Chemistry*. 2019;1(3):282-291. <https://doi.org/10.1016/j.trechm.2019.02.007>
4. Tom G, Schmid SP, Baird SG, Cao Y, Darvish K, Hao H, et al. Self-driving laboratories for chemistry and materials science. *Chemical Reviews*. 2024;124(16):9633-9732. <https://doi.org/10.1021/acs.chemrev.4c00055>
5. Burger B, Maffettone PM, Gusev VV, Aitchison CM, Bai Y, Wang X, et al. A mobile robotic chemist. *Nature*. 2020;583(7815):237-241. <https://doi.org/10.1038/s41586-020-2442-2>
6. MacLeod BP, Parlane FGL, Morrissey TD, Häse F, Roch LM, Dettelbach KE, et al. Self-driving laboratory for accelerated discovery of thin-film materials. *Science Advances*. 2020;6(20):eaaz8867. <https://doi.org/10.1126/sciadv.aaz8867>
7. Szymanski NJ, Rendy B, Fei Y, Kumar RE, He T, Milsted D, et al. An autonomous laboratory for the accelerated synthesis of inorganic materials. *Nature*. 2023;624(7990):86-91. <https://doi.org/10.1038/s41586-023-06734-w>
8. Stach E, DeCost B, Kusne AG, Hattrick-Simpers J, Brown KA, Reyes KG, et al. Autonomous experimentation systems for materials development: A community perspective. *Matter*. 2021;4(9):2702-2726. <https://doi.org/10.1016/j.matt.2021.06.036>
9. Rapp JT, Bremer BJ, Romero PA. Self-driving laboratories to autonomously navigate the protein

- fitness landscape. *Nature Chemical Engineering*. 2024;1(1):97-107.
<https://doi.org/10.1038/s44286-023-00002-4>
10. Roch LM, Häse F, Kreisbeck C, Tamayo-Mendoza T, Yunker LPE, Hein JE, et al. ChemOS: An orchestration software to democratize autonomous discovery. *PLOS ONE*. 2020;15(4):e0229862.
<https://doi.org/10.1371/journal.pone.0229862>
 11. Sim M, Vakili MG, Strieth-Kalthoff F, Hao H, Hickman RJ, Miret S, et al. ChemOS 2.0: An orchestration architecture for chemical self-driving laboratories. *Matter*. 2024;7(9):2959-2977.
<https://doi.org/10.1016/j.matt.2024.04.022>
 12. Fei Y, Rendy B, Kumar R, Dartsi O, Sahasrabudhe HP, McDermott MJ, et al. AlabOS: a Python-based reconfigurable workflow management framework for autonomous laboratories. *Digital Discovery*. 2024;3(11):2275-2288. <https://doi.org/10.1039/d4dd00129j>
 13. Bai J, Cao L, Mosbach S, Akroyd J, Lapkin AA, Kraft M. From platform to knowledge graph: Evolution of laboratory automation. *JACS Au*. 2022;2(2):292-309.
<https://doi.org/10.1021/jacsau.1c00438>
 14. Boiko DA, MacKnight R, Kline B, Gomes G. Autonomous chemical research with large language models. *Nature*. 2023;624(7992):570-578. <https://doi.org/10.1038/s41586-023-06792-0>
 15. Carbonell P, Jervis AJ, Robinson CJ, Yan CY, Dunstan M, Swainston N, et al. An automated Design-Build-Test-Learn pipeline for enhanced microbial production of fine chemicals. *Communications Biology*. 2018;1:66. <https://doi.org/10.1038/s42003-018-0076-9>
 16. Hillson N, Caddick M, Cai YZ, Carrasco JA, Chang MW, Curach NC, et al. Building a global alliance of biofoundries. *Nature Communications*. 2019;10(1):2040.
<https://doi.org/10.1038/s41467-019-10079-2>
 17. Holland I, Davies JA. Automation in the life science research laboratory. *Frontiers in Bioengineering and Biotechnology*. 2020;8:571777. <https://doi.org/10.3389/fbioe.2020.571777>
 18. Launching the Genesis Mission, Exec. Order No. 14363, 90 Fed. Reg. 55035 (Nov. 28, 2025).
<https://www.federalregister.gov/d/2025-21665>
 19. Ministry of Industry and Information Technology of the People's Republic of China, et al. Notice on issuing the 15th Five-Year Plan for the Development of the Pharmaceutical Industry. MIIT Joint Planning [2026] No. 210 (Aug. 28, 2026). [In Chinese].
https://www.miit.gov.cn/zwgk/zcwj/wjfb/tz/art/2026/art_9463adac4db040159f6cbf482675eeea.html
 20. Joint Committee for Guides in Metrology. International vocabulary of metrology-Basic and general concepts and associated terms (VIM), 3rd ed. JCGM 200:2012. 2012.
<https://doi.org/10.59161/jcgm200-2012>
 21. Joint Committee for Guides in Metrology. Evaluation of measurement data-Guide to the expression of uncertainty in measurement. JCGM 100:2008. 2008.
<https://doi.org/10.59161/jcgm100-2008e>
 22. Joint Committee for Guides in Metrology. Evaluation of measurement data-The role of measurement uncertainty in conformity assessment. JCGM 106:2012. 2012.
<https://doi.org/10.59161/jcgm106-2012>
 23. International Organization for Standardization. Automated liquid handling systems-Parts 1-3: ISO 23783-1:2022, ISO 23783-2:2022, and ISO 23783-3:2022. Geneva: ISO; 2022.
 24. International Organization for Standardization. Piston-operated volumetric apparatus-Part 8: Photometric reference measurement procedure for the determination of volume. ISO 8655-8:2022. Geneva: ISO; 2022.

25. International Organization for Standardization, International Electrotechnical Commission. General requirements for the competence of testing and calibration laboratories. ISO/IEC 17025:2017. Geneva: ISO; 2017.
26. Gadelrab MS, Abouhogail RA. Towards a new generation of digital calibration certificate: Analysis and survey. *Measurement*. 2021;181:109611. <https://doi.org/10.1016/j.measurement.2021.109611>
27. Hackel S, Schönhals S, Doering L, Engel T, Baumfalk R. The digital calibration certificate (DCC) for an end-to-end digital quality infrastructure for Industry 4.0. *Sci*. 2023;5(1):11. <https://doi.org/10.3390/sci5010011>
26. Hobbs KL, Mote ML, Abate MCL, Coogan SD, Feron EM. Runtime assurance for safety-critical systems: An introduction to safety filtering approaches for complex control systems. *IEEE Control Systems*. 2023;43(2):28-65. <https://doi.org/10.1109/MCS.2023.3234380>
27. Hentz NG, Bradshaw JT. Guide to liquid volume measurements: A review of methods and technologies. *SLAS Technology*. 2026;36:100382. <https://doi.org/10.1016/j.slant.2025.100382>
30. Torres-Acosta MA, Lye GJ, Dikicioglu D. Automated liquid-handling operations for robust, resilient, and efficient bio-based laboratory practices. *Biochemical Engineering Journal*. 2022;188:108713. <https://doi.org/10.1016/j.bej.2022.108713>
31. Wilkinson MD, Dumontier M, Aalbersberg IJJ, Appleton G, Axton M, Baak A, et al. The FAIR Guiding Principles for scientific data management and stewardship. *Scientific Data*. 2016;3:160018. <https://doi.org/10.1038/sdata.2016.18>
32. Salazar-Villacis P, Benyahia B. The ADePT framework for assessing autonomous laboratory robotics. *Communications Chemistry*. 2026;9(1):99. <https://doi.org/10.1038/s42004-026-01932-9>
33. Reder GK, Gower AH, Kronström F, Halle R, Mahamuni V, Patel A, et al. Genesis-DB: a database for autonomous laboratory systems. *Bioinformatics Advances*. 2023;3(1):vbad102. <https://doi.org/10.1093/bioadv/vbad102>