

Cite this: DOI: 00.0000/xxxxxxxxxx

AEGIS: Assay-Aware Protocol Validation and Runtime Monitoring for Open-Source Liquid Handling Robots[†]Priyanka V. Setty,^{*a,b} Arvind Ramanathan,^a Ian Foster,^{a,b} and Rick Stevens^{a,b}

Received Date

Accepted Date

DOI: 00.0000/xxxxxxxxxx

Self-driving laboratories increasingly rely on low-cost liquid handling robots such as the Open-trons OT-2, but these platforms ship without the pressure-based aspiration monitoring found on Hamilton or Tecan systems and are typically operated open-loop. Two complementary failure modes go undetected: protocols that are syntactically valid but violate assay-specific invariants (for example, tip reuse between PCR template and a no-template control), and physical execution failures (partial dispense, air bubbles, missing tips) that occur at runtime. We present AEGIS, a two-layer guardian that addresses both. Layer 1 combines a curated machine-readable assay rule database with a large language model that reasons over OT-2 Python code, reaching an adjusted F_1 of 0.97 on a 24-protocol benchmark spanning five assay families and outperforming both rules-only and LLM-only ablations across five backends; a free open-weight model ties the best proprietary one, so the approach does not depend on a paid API. Layer 2 fits a principal-component world model to YOLO-cropped four-frame pipette trajectories and detects physical anomalies on the p1000 channel; under a leakage-free leave-one-plate-out evaluation it reaches an average precision of 0.89 and an operating-point F_1 of 0.71 (AUROC 0.80), a deployment-faithful figure that matches the live demonstration, and we characterize the small-pipette (p20) resolution limit at which the front view degrades ($F_1=0.47$). A proof-of-concept live demonstration on a physical OT-2 with a front-view camera (five replicates per condition) catches the planted no-tip failure deterministically (5/5) and partial dispense under the default cascade on red and yellow dye (3/5 each); an always-VLM operating point that uses a three-assessment self-vote prompt lifts partial-dispense recall to 5/5 on both red and yellow. Water remains a principled limit of any front-view-only system (cascade 0/5, always-VLM 1/5 at low confidence), which AEGIS surfaces in the VLM's own low-confidence reasoning rather than over-committing to a verdict. End-to-end the OT-2 halts at the next protocol-level boundary at an effective-halt latency of 13.27 ± 0.47 s ($n=9$ paused runs), and the cascade triage holds projected per-plate VLM cost at \$1.63 versus \$10.33 for an always-VLM baseline. AEGIS is open source and is, to our knowledge, the first system to unify pre-flight assay-aware validation with runtime visual monitoring for an open-source liquid handler.

1 Introduction

Self-driving laboratories (SDLs) are reshaping experimental science by closing the design-build-test-learn loop with autonomous execution: predictive models propose experiments, robotic platforms run them, and the resulting data updates the model in a continuous loop, ideally without human intervention.^{1–3} The scientific integrity of this loop depends on whether each wet-lab step actually does what the protocol intended; at scale, no human can audit every well to verify. A silently failed

wet-lab run does not produce an error; it produces *corrupted data that compounds* across subsequent model updates and experimental suggestions. Yet across the SDL stack, the robots themselves are *biologically blind*: a liquid handler executes whatever Python or vendor-specific protocol it is given, with no awareness of assay-level intent. Two distinct failure modes can silently corrupt a run (Fig. 1): assay-level violations that survive every syntactic check, and physical execution failures that the robot's own log will not flag. The A-Lab² synthesized 41 of 58 target compounds (a 71% success rate), a vivid reminder that “the robot ran” is not the same as “the experiment worked.”

^a Data Science and Learning Division, Argonne National Laboratory, Lemont, IL 60439, USA. E-mail: psetty@anl.gov

^b Department of Computer Science, University of Chicago, Chicago, IL 60637, USA.

[†] Electronic Supplementary Information (ESI) available: rule database, per-protocol breakdowns, fleet scaling logs, demo run videos. See DOI: 00.0000/00000000.

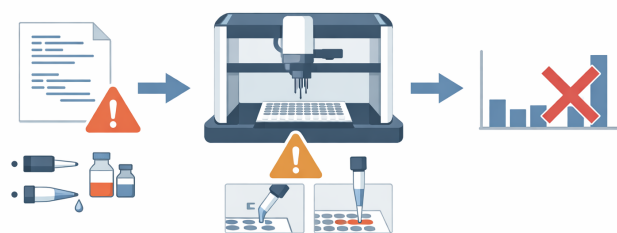


Fig.1 AEGIS guards against two silent failure modes that corrupt autonomous-lab data: assay-level (semantic) errors in the protocol code, and physical execution failures inside the liquid handler during the run. Both propagate as bad data into downstream analyses and the predictive models trained on them.

1.1 Assay-level (semantic) errors

An assay-aware reading of the protocol's Python source catches errors that no syntactic check can: the tip touching a no-template control should never have touched a PCR template,⁴ standards in a Bradford protein assay⁵ must be dispensed low-to-high to limit carryover, and in a high-throughput drug-screening assay⁶ a control well must never receive the drug under test. A protocol can be syntactically valid, accepted by the vendor's simulator, and executed without runtime errors, yet still corrupt every downstream measurement through such an assay-level (semantic) violation.

A wave of LLM agents now converts natural-language experimental intent into executable protocols,^{3,7,8} but none of them enforces the per-assay invariants a wet-lab scientist would apply on a code read. Coscientist couples GPT-4 with web search and OT-2 control, correcting its own coding errors by consulting the equipment manual at runtime; the error handling is reactive code correction, not pre-flight assay validation.³ LabscripAI generates liquid-handling scripts from natural language across multiple platforms including the OT-2, and was used in production for 298 GFP variants across 53 teams, but its 55-task benchmark measures protocol-generation success and executability, not assay-specific rules.⁷ Opentrons' commercial AI tool pairs a retrieval-augmented generator with the Protocol Designer to produce executable code, also without assay-level validation.⁸ Adjacent work (ChemCrow's tool-augmented chemistry agent with built-in safety screening,⁹ ORGANA's GPT-4-driven robotic chemistry assistant that turns natural-language goals into syntactically verified robot plans with human-in-the-loop troubleshooting,¹⁰ CLAIRify's verifier-assisted translation of natural-language protocols into an executable robot description language,¹¹ and BioPlanner's benchmark of LLM protocol planning in biology¹²) each advances protocol authoring or safety screening, but none encodes the per-assay biology invariants (fresh tips between PCR template and NTC, low-to-high standards in Bradford, mix steps after each serial dilution) that distinguish a correct biology protocol from

a merely-compiling one. The closest formal-methods precedent is BioScript, a domain-specific language that statically verifies against unsafe chemical mixing on microfluidic laboratories-on-a-chip,¹³ but its scope is microfluidics, not bench-scale liquid handling. The unaddressed gap across this body of work is consistent: existing systems check that a protocol is syntactically and logically well-formed, but none reads it through the lens of assay-specific biology invariants.

PRISM, a pre-flight planning and validation system, checks the biological plausibility of a proposed protocol and simulates inter-robot plate transfers in an Omniverse-based digital twin before execution.¹⁴ AEGIS differs along two axes. First, where PRISM judges biological plausibility at the level of the overall protocol, Layer 1 checks biological correctness against *explicit, assay-specific rules*, the finer and auditable per-assay invariants described above (for example, a tip carried from a PCR template into a no-template control). Second, the digital twin captures the *physical movements* of the workflow, gantry motion and plate transfers between instruments, but not the liquid handler's internal pipetting; aspiration and dispensing inside the tip lie outside the simulated scene, and the observational window ends at the start of the run, the runtime gap Layer 2 fills.

1.2 Physical execution errors

Even a correct protocol can fail at runtime through purely physical events: a missing tip, an air bubble drawn during aspirate, accumulated residual liquid in a reused tip, or a missed dispense. These failures are invisible to the protocol code and require runtime observation of the robot itself.

At runtime, industrial liquid handlers from Hamilton and Tecan ship with pressure-based monitoring.^{15,16} Hamilton's Total Aspiration and Dispense Monitoring (TADM) records a pressure-versus-time trace for every pipetting channel and compares it against expected per-liquid-class envelopes. Tecan combines capacitive (cLLD) and pressure-based (pLLD) liquid-level detection with Process Monitored Pipetting for similar coverage. These capabilities ship only on platforms in the \$100,000–\$300,000 class with proprietary scripting and orchestration software (Hamilton Venus, Tecan Evoware), an integration profile incompatible with the open, Python-native, multi-robot scenarios most academic SDLs target. The Opentrons OT-2, the most widely deployed open-source liquid handler in academic SDLs,^{7,8,14} ships with no aspiration or dispense monitoring, no liquid-level detection, and no error reporting beyond Python exceptions. Any runtime quality control for the OT-2 must therefore be added externally, typically through cameras or auxiliary sensors.

Vision-based monitoring for the OT-2 has so far focused on narrow detection tasks. A YOLOv8 pipeline has been deployed on the OT-2 to detect pipette tips and liquid volumes in real time, with the authors explicitly noting inability to detect air bubbles as future work.¹⁷ An attention-augmented YOLOv8 variant targets liquid retention (residual droplets) on tips across different liquid colours and backgrounds, but is restricted to per-frame classification of a single failure type rather than analyzing the trajectory across the pipetting cycle.¹⁸ The LIRA edge module uses

vision-language models for localisation and post-action inspection in SDLs and reports a $\sim 10\times$ reduction in localisation time, but is fundamentally reactive: it inspects *after* each action without a model of what the protocol expected to happen, treating every step as an independent inspection task with no shared prior across the run.¹⁹

A separate body of work uses learned world models as the basis for anomaly detection. The prediction error of a model-based RL world model has been shown to serve as a domain-agnostic anomaly signal on locomotion and flight tasks.²⁰ Hybrid frameworks combine supervised failure classification with world-model anomaly thresholds for visual inspection of industrial gauges.²¹ The dominant world-model architectures (DreamerV3,²² IRIS,²³ TD-MPC2²⁴) are evaluated on games, locomotion, or tabletop manipulation. MATTERIX provides a GPU-accelerated digital twin of chemistry lab automation built on Isaac Sim/Lab,²⁵ but its purpose is in-silico workflow testing rather than runtime anomaly detection. Methodologically the closest precedent is industrial visual anomaly detection (MVTec AD,²⁶ PatchCore²⁷), which trains on normal images alone and flags deviations. None of these world-model or train-on-normal anomaly approaches has been applied to liquid handling, which is the gap AEGIS Layer 2 closes by adapting the train-on-normal paradigm to temporal pipette trajectories on the OT-2.

A parallel and faster-moving line scales foundation and vision-language-action (VLA) models to laboratory and embodied automation. RoboChemist couples vision-language models with a robotic agent to carry out multi-step chemistry protocols while monitoring compliance with experimental norms,²⁸ and recent surveys catalogue how foundation models are reshaping both high-level task planning and low-level control for robot manipulation, together with their scalability and safety challenges.²⁹ These systems learn rich, general policies from large datasets and substantial compute. AEGIS deliberately takes the opposite, lightweight and data-efficient route appropriate to a proof of concept: a train-on-normal anomaly monitor and an explicit rule checker that need no large training corpus and run on commodity hardware. We see the foundation-model path as the natural scale-up, a fine-tuning target once monitoring datasets grow large enough to cover the failure modes AEGIS does not yet handle, and we return to it in the future-work discussion (Sec. 4).

1.3 Contributions and goals

We address both gaps on the Opentrons OT-2. The OT-2 exposes its protocols as Python and integrates easily with multi-robot SDL workflows through its open HTTP API, but ships with no built-in tip-level or pressure-based sensing. We present **AEGIS** (Automated Experiment Guardian and Inspection System), a proof-of-concept two-layer guardian that runs end-to-end on a physical OT-2 with off-the-shelf hardware: a front-view USB camera, a laptop, and the existing Opentrons API.

Layer 1 is a pre-flight assay-aware checker that reads the protocol's Python source and flags *violations of a curated 22-rule database* of assay-specific invariants spanning five families (contamination, wrong well order, missing or out-of-order steps). We

frame its evaluation as a *rule-coverage benchmark*, whether a violation of each formalized rule is detected, rather than a claim of detecting arbitrary or unseen semantic errors, and we assess both frontier and freely available open models as the underlying reasoner.

Layer 2 is a runtime monitor that captures the pipette trajectory from the front-view camera, detects silent physical failures (partial dispense, no aspirate, no tip) as they occur, and pauses the OT-2 via its API to notify the operator; it adapts the train-on-normal anomaly-detection paradigm from industrial visual inspection, using a lightweight PCA reconstruction over temporal four-frame pipette trajectories. Because its signal is the tip's geometric trajectory rather than reagent appearance, detection does not depend on reagent colour; truly transparent fluids such as water are the exception, a physical limit of a front-view monitor that we report honestly (Section 3.3). The PCA scorer is a deliberately simple, data-efficient baseline for the small-data regime of a proof of concept, designed to scale as deployments accumulate trajectories (Section 4).

We demonstrate AEGIS end-to-end on a physical OT-2 across multiple failure modes, characterizing where the system catches reliably, where it currently falls short, and what extensions would close the remaining gaps. We release the validators, rule database, trained model weights, captured datasets, and demonstration protocols under an open-source license.

2 Methods

AEGIS is a two-layer guardian for the Opentrons OT-2 (Fig. 2): a pre-flight assay-aware validator (Layer 1) that reads the protocol's Python source before the run, and a runtime visual monitor (Layer 2) that watches the pipette through a camera, scores per-well trajectories with a PCA world model trained on normal trajectory images, and routes ambiguous wells to a vision-language model (VLM) that arbitrates pause decisions. Both layers run as a single per-robot agent and can be replicated across a fleet without code changes; the multi-robot scaling experiment is reported in the supplementary information (SI Section A). Source code, weights, and datasets are released under the MIT license (see Conclusions).

2.1 Protocol benchmark and failure injection

We curated 24 OT-2 Python protocols spanning five assay families: serial dilution, polymerase chain reaction (PCR), Bradford protein assay, cell-treatment with drug controls, and generic transfer/normalization. Of these, 11 are correct (drawn from Opentrons' official tutorials and the Protocol Library, with minor formatting cleanup) and 13 are buggy variants in which an assay-relevant error has been injected: a shared tip between PCR template and the no-template control, a missing master-mix step, an inverted serial-dilution direction, a high-to-low Bradford standards series, and similar. Each buggy protocol tests exactly one rule, simplifying per-rule evaluation. A separate set of three multi-bug protocols (each with two or three concurrent errors) is used to test whether the validator surfaces multiple independent violations from a single code read. We feed every protocol

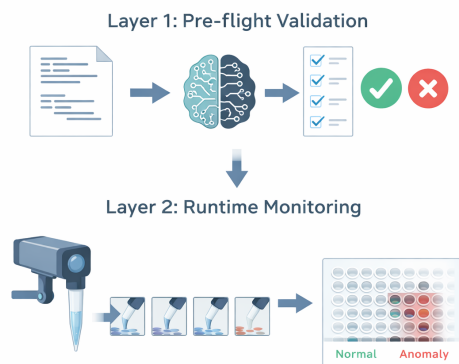


Fig. 2 AEGIS two-layer architecture. Layer 1 reads the OT-2 protocol’s Python source before the run and flags assay-level violations against a curated rule database. Layer 2 watches the pipette through a camera during execution, scores per-well trajectories with a PCA world model trained on normal trajectory images, and routes ambiguous wells to a vision-language model that arbitrates the pause decision.

through Layer 1 and compare the validator’s predicted pass/fail and per-rule call against the injected ground truth (Section 3.1). Two of the eleven nominally-correct tutorial protocols in fact contain a genuine tip-reuse bug; we therefore report both a strict score against the original labels and an adjusted score that relates these two as buggy, crediting the models that flag them.

2.2 Layer 1: assay rules and validator

We formalized 22 assay-specific rules in a machine-readable JSON schema. Each rule has an identifier (e.g. PCR-2: “fresh tip between every PCR template and the NTC”), a natural-language statement, a severity level, an assay scope, and optional structural indicators the LLM uses during code inspection. Rules cover contamination (tip-reuse patterns), ordering (low-to-high standards, dilution direction), completeness (master-mix presence, mixing after each dilution step), and source/destination correctness (drug only in treated wells, diluent from the correct reservoir). The rule database is extensible by design: adding an assay family requires only editing the rules file, with no code changes to the validator.

Layer 1 itself is a single-pass LLM call that consumes the rule database, a curated Opentrons API reference describing the semantics of `transfer`, `distribute`, `consolidate`, and the `new_tip` parameter, and the protocol’s Python source. We supply this Opentrons API reference in-context rather than relying on each backend’s pretrained knowledge of the library, so the comparison reflects assay-level reasoning rather than familiarity with Opentrons syntax and does not disadvantage the open-weight model. The LLM is asked in one structured prompt for (i) the inferred assay family with a confidence label, (ii) a rule-by-rule evaluation that lists offending wells and cites each violated rule, and (iii) a final pass/fail verdict. We evaluate five backends spanning capability tiers: OpenAI o4-mini (reasoning), Claude Opus 4 and GPT-4o (frontier), Claude Sonnet 4.5 (mid-tier), and

NVIDIA Nemotron-3 Ultra 550B (open-weight, served through NVIDIA NIM). The open-weight model is included deliberately to test whether assay-aware validation depends on a paid frontier API or can run on openly available weights, a deployment-cost question for self-driving laboratories. All backends are scored on the same 24 single-bug protocols; the multi-bug set is a separate test reported below. Combining classification and rule checking in a single call lets the model reason about cross-step context, for example that a loop’s `transfer` call inherits the `new_tip` default, which a staged pipeline would lose. The design is deliberately hybrid: rules tell the LLM *what* to check; the LLM figures out *how* to check it in the code. To isolate each component’s contribution we evaluate Layer 1 in three configurations: the full hybrid (rules + LLM), an LLM-only ablation (the same prompt and Opentrons API reference, with only the rule database withheld, so the ablation measures the rules’ contribution and not library knowledge), and a deterministic rules-only baseline (structural pattern-matching without the LLM). For the open-model comparison we also record per-protocol wall-clock latency for each backend.

2.3 Layer 2 setup and data collection

A Logitech webcam mounted on a stand pointed at the OT-2 deck captures the work area at 1 fps, with a plain black cardboard panel placed behind the deck to simplify tip segmentation against a uniform background. The live monitor reads the Opentrons HTTP API to obtain each pipetting command’s completion timestamp and aligns the corresponding captured frame; alignment is verified to within ± 1 frame. Cross-clock drift between the OT-2 and the workstation is auto-detected at run start from the HTTP response Date header (typically ~ 25 s in our setup) and applied to every command timestamp before frame mapping.

For the offline evaluation we collected 64 p1000 well trajectories on the same hardware: 26 physically-normal wells (1–2 from the early wells of each plate) and 38 planted failures, across three reagent contrasts (red, yellow, water) and four failure types (partial dispense, no dispense, no tip, air bubble). Each trajectory consists of four frames extracted at protocol-level keypoints: `ASPIRATE`, `TRANSIT`, `DISPENSE`, and `POST-DISPENSE`. Two `NO_TIP` samples are excluded from the quantitative evaluation because YOLO returns no detection on blank tips and the live cascade instead catches those via the separate `BLANK_FLOOR` rule (Section 2.6), leaving 62 samples for the evaluation reported in Section 3.2.

We report a leakage-free leave-one-plate-out (LOPO) protocol as the canonical evaluation, because it mirrors deployment on a previously unseen plate. For each held-out plate, the PCA world model (Section 2.5) is fit on the normal trajectories of *all other* plates, the anomaly threshold is calibrated from the held-out plate’s own first normal well, and every remaining well of the held-out plate is then scored. No plate ever contributes both to the manifold it is scored against and to its own threshold, so the reported numbers reflect transfer to an unseen plate rather than in-sample reconstruction.

The primary evaluation uses the p1000 single-channel pipette. To probe the resolution limit of the camera, we additionally col-

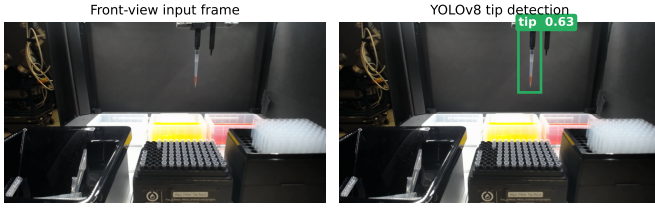


Fig. 3 YOLOv8-nano tip detection on a representative camera frame. Left: raw camera input. Right: the fine-tuned detector returns a tight bounding box around the pipette tip (confidence shown above the box). The detected region is cropped, padded, and resized to the fixed canvas used by the PCA world model.

lected a matched p20 trajectory set on the same hardware and ran the identical LOPO protocol; the p20 result quantifies where the current camera resolution stops resolving the smaller tip and fluid column (Section 3.2). Extending the pipeline to the 8-channel multichannel head is in progress and reported as future work (Section 4).

2.4 Tip detection (YOLOv8)

To make the per-well pipette region available for downstream modelling, YOLOv8-nano was fine-tuned to localise the pipette tip, on 69 hand-labelled frames in which the tip bounding box was annotated (including tipless frames, where the correct output is no detection). Training took ~ 2 minutes on a single GPU and reached mAP50 of 0.995 on a held-out split. The detector performs geometric tip localisation only and is agnostic to whether a well is normal or failing; interpreting the cropped trajectory is left to the PCA world model and the VLM. At inference, every captured frame yields a tight tip bounding box (Fig. 3) that is cropped, padded, and resized to a fixed canvas before being passed to the world model.

2.5 PCA trajectory world model

For each well, AEGIS concatenates the YOLO-cropped tip patches from the four keyframes (greyscale, 48×48 each) into one 9,216-dimensional trajectory vector. A principal-component analysis with $k=15$ components, retaining 97.7% of variance, is fit on the normal trajectories; k was selected once on the full set of 26 normals, and under the leave-one-plate-out evaluation (Section 2.3) the model is refit per fold. The number of components was chosen at the variance-explained knee, beyond which the reconstruction-error threshold collapses to floating-point noise and the detector degenerates (Section 3.2, Fig. 13). Reconstruction error $\|x - \hat{x}\|_2^2$ serves as the anomaly score. Alternative anomaly detectors were evaluated head-to-head against PCA on the same corpus and underperformed at this data scale: an uncalibrated VLM, an Isolation Forest, and a one-class SVM all over-flag the normal wells, and a dense autoencoder learns a near-identity reconstruction at this sample size. The choice of linear PCA is discussed in Results (Section 3.2) and Section 4.

2.6 Real-time deployment: cascade and always-VLM

The reconstruction-error scale drifts by roughly a factor of two between sessions due to lighting, camera profile, and ambient conditions. Rather than re-fit PCA or retune a global threshold, AEGIS calibrates the cascade per run on the first six wells, which by design carry standards, controls, or normalisation reagents that are physically normal. We record their PCA scores $c_1, \dots, c_6$ and define the silent-pass band as $\text{median}(c_i) \pm 0.7 \cdot \text{MAD}(c_i)$, where MAD is the median absolute deviation. Both the median and the MAD are themselves median operations, so the band is robust to a single anomalous calibration well, and the per-run anchor absorbs cross-session drift without retraining.

For each post-calibration well, the cascade emits one of three routing decisions (Fig. 4):

- **SILENT-PASS** when the PCA score lies inside the per-run calibration band. No API call is issued.
- **BLANK FLOOR** when the score is below 10^{-4} , signalling that the YOLO detector produced no tip-containing crop in one or more frames. This is the dominant signature of missing-tip and gross-occlusion failures and is routed to the VLM to confirm.
- **GRAY ZONE** for any other score outside the calibration band. The trajectory is routed to the VLM with the YOLO-cropped trajectory frames.

The PCA scorer is never permitted to trigger a robot pause on its own. The VLM acts as a second-stage validator that confirms or rejects each PCA-flagged well, and is the sole arbiter of any pause decision: the PCA cascade triages, the VLM decides. This design eliminates a class of false positives in which an unusual but valid well would otherwise halt a working run.

Wells routed by the cascade enter a Claude Opus 4.7 vision call that receives the four cropped trajectory frames together with a reagent-colour-aware system prompt. The prompt is structured as three in-call assessments by distinct lenses (SKEPTICAL, ROUTINE, and EXPERT). Each lens independently returns {PASS, FAIL} with a confidence and a failure-type label drawn from {partial dispense, no aspirate, no tip}. The aggregator applies an ANY-FAIL rule: if any lens returns FAIL, the overall verdict is FAIL. The reagent-colour parameter enters the prompt explicitly because Claude’s prior on coloured fluid presence is strong, and a colour-aware lens is needed to avoid hallucinating “red fluid” on an empty tip from a water run.

AEGIS also exposes an always-VLM flag as a second operating point: rather than letting PCA triage, it bypasses the silent-pass band and forces every post-calibration well through the VLM gate. This is a more exhaustive check that always invokes the VLM, trading per-plate cost (Section 3.3) for coverage; it is the safety-maximising setting for cases where one does not want to rely on PCA triage. We stress that always-VLM is not a fix for transparent reagents: water fails in both operating modes (Section 3.3), because the limitation there is one of image resolution rather than of the routing policy, the consumer Logitech webcam places too few pixels on the tip to resolve a transparent meniscus. We therefore treat the two modes only as different points on

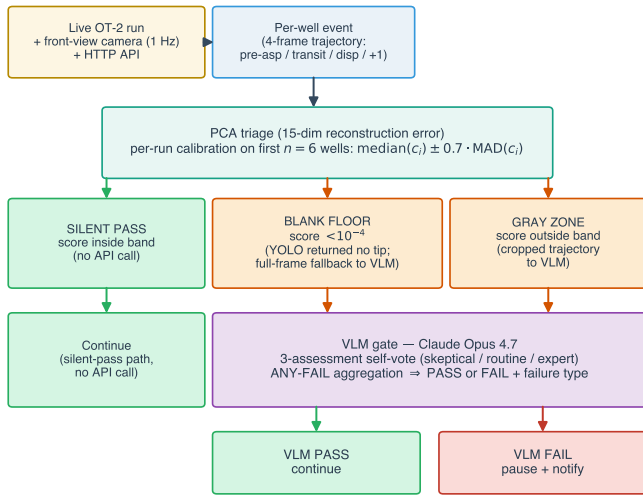


Fig. 4 Layer 2 live cascade. Per-well events flow from the camera and the OT-2 HTTP API into PCA triage with three routing zones (SILENT_PASS inside the per-run calibration band, BLANK_FLOOR for blank-tip cases, GRAY_ZONE otherwise). Wells outside the silent-pass band, along with all wells in always-VLM mode, are routed to a 3-assessment self-vote VLM gate that returns the sole pause verdict.

a cost-coverage curve, and defer the transparent-fluid resolution limit and its potential fixes to Section 4.

For the live demonstration the per-robot agent acts on the cascade’s pause verdict: when the VLM gate returns FAIL, the agent issues a pause via the Opentrons HTTP API and the OT-2 halts at the next protocol-level boundary (the drop-tip command at the end of the failed well in our demo protocols). Layered notifications (terminal log, macOS desktop notification, optional Pushover webhook) carry the well identifier, the VLM failure-type label, the lens-by-lens reasoning excerpts, and the path to the saved trajectory frames. Demo protocols exercise the cascade end-to-end on a physical OT-2 with a p1000 single-channel pipette across three physical-failure types (partial dispense, no aspirate, no tip) and three reagent contrasts (red dye, yellow dye, water); a fourth mode, air bubble, could not be tested live because reliably injecting a visible air gap into a running dispense proved difficult, so it is evaluated only in the offline set (Section 3.2). Each injects a single planted failure at a designated transfer and drops the tip after each well so that the cascade’s pause arrives before contamination of the next destination. Each condition is run for five consecutive replicates on the same physical OT-2 with a fixed detector configuration, and we report recall as $X/5$ with a Wilson 95% confidence interval (false pauses are reported separately as raw counts, being uninformative as a proportion at $N=5$). No run was excluded on the basis of its outcome.

3 Results and Discussion

We report Layer 1 protocol-validation accuracy and ablations (Section 3.1), Layer 2 runtime-monitoring accuracy with baselines and cross-validation (Section 3.2), and the live end-to-end demonstration on a physical OT-2 (Section 3.3). The two layers cover non-overlapping failure surfaces: Layer 1 catches code-level failures invisible to cameras, Layer 2 catches physical-execution

failures unpredictable from code, and neither can substitute for the other; AEGIS therefore runs both as separate layers rather than fusing them.

3.1 Layer 1: protocol validation

All five LLM backends (o4-mini, the open-weight NVIDIA Nemotron-3 Ultra, Claude Opus 4, GPT-4o, and Claude Sonnet 4.5) classify the assay family correctly on all 24 benchmark protocols (24/24); classification is essentially saturated, with every protocol identified with high confidence from a combination of code structure, variable names, and inline comments.

Every LLM-backed configuration also achieves 100% recall on the 13 injected bugs in the single-bug benchmark: no failure mode in our protocol set evades detection. The five systems differ only in their false-positive rate on the 11 nominally correct protocols. Two of those “correct” protocols are Opentrons-tutorial serial-dilution scripts that actually contain a real tip-reuse bug every model legitimately flags; under the adjusted scoring that credits those flags, and with every backend scored on the same 24 single-bug protocols, the five hybrid configurations cluster tightly: o4-mini, the open-weight Nemotron, and Claude Opus 4 all reach $F_1=0.968$ (a single false positive each), GPT-4o 0.938 (two false positives), and Claude Sonnet 4.5 0.909 (three), against a deterministic rules-only baseline of 0.741 (Fig. 5; 95% bootstrap confidence intervals span roughly $[0.88, 1.00]$ for the hybrids). The robust observation is that with the rule database present every backend lands within one to three false positives of perfect, regardless of scale, vendor, or cost. The open-model comparison headline follows directly: assay-aware validation does not depend on a paid frontier API, the free open-weight Nemotron ties the best proprietary backend at $F_1=0.968$ with 100% bug recall. The open model’s practical price is latency rather than accuracy, roughly $9\times$ slower per protocol than GPT-4o on the free hosted tier (~ 57 s versus ~ 6 s), which is acceptable for a pre-flight check that runs once before a multi-hour run.

Stripping the rules exposes each model’s intrinsic behaviour, and here the backends diverge sharply. Recall stays at 100% for all, so once again the difference is entirely false positives, but without the rule database to bound what to check the count jumps from one–three (hybrid) to six–eight. o4-mini falls to $F_1=0.800$ (six false positives), and GPT-4o, the open Nemotron, and Sonnet 4.5 to 0.789 (eight); only Claude Opus 4 retains a strong domain prior at 0.909 (three). Without rules GPT-4o flags ten of the eleven correct protocols, essentially refusing to certify anything. Running deterministic regex pattern-matching rules alone (no LLM) misses five bugs, four that require code-flow reasoning (e.g. tip reuse through a loop with a default `new_tip` parameter, source-reservoir mismatches across function boundaries) and one tutorial tip-reuse variant whose structural pattern the regex does not match, landing the rules-only baseline at 0.741. Every hybrid configuration in Fig. 5 therefore dominates both its own LLM-only ablation and the rules-only baseline.

The false-positive structure carries the main lessons of the Layer 1 comparison. First, the LLM is never the detection bottleneck: every backend catches all injected bugs, so it is the rule

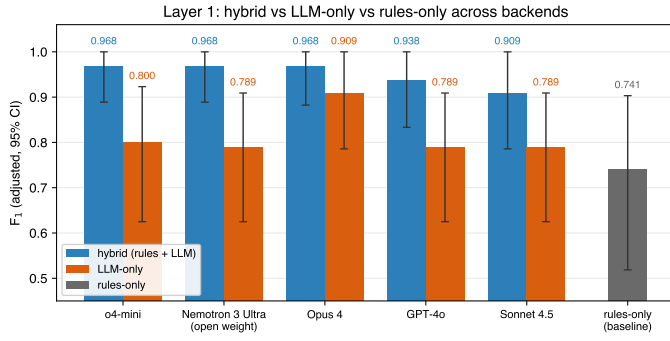


Fig. 5 Layer 1 ablation across five backends spanning capability tiers (o4-mini reasoning; frontier Opus 4 and GPT-4o; the mid-tier Sonnet 4.5; and the open-weight Nemotron-3 Ultra) plus the deterministic rules-only baseline, all scored on the same 24 single-bug protocols with 95% bootstrap confidence intervals on the adjusted F_1 . Each backend shows its hybrid (rules + LLM) and LLM-only bars; rules-only is the grey bar at 0.741. With rules, o4-mini, Opus 4, and the open-weight Nemotron tie at $F_1=0.968$ (one false positive each); GPT-4o and Sonnet 4.5 add one and two more. All configurations achieve 100% recall, so the entire spread is false-positive suppression, and it is supplied by the rules rather than the model.

database, not the model, that supplies the precision contract. Second, once the rules are present, model scale, vendor, and cost are largely irrelevant, a free open-weight model ties the best proprietary frontier backend, which is the practical result for cost-constrained self-driving laboratories. Third, a larger or newer model is not automatically more precise: Sonnet 4.5 is in fact the weakest hybrid, over-flagging more than Opus, o4-mini, and even the open-weight Nemotron, both with and without rules, so the route to fewer false positives is the rule scaffold rather than a bigger reasoner. Finally, Layer 1’s remaining errors are harmless: it occasionally over-flags correct code, but never misses a real bug, which is the safe way for a safety checker to be wrong.

To test whether Layer 1 reports every violation it finds rather than only a pass/fail verdict, we separately evaluated Claude Opus hybrid on three multi-bug protocols carrying 2–3 concurrent injected bugs each. AEGIS detected all 8/8 injected violations and additionally surfaced one legitimate downstream contamination that the injected bugs would cause, with correct per-violation citations in every case.

After the adjustment, Opus’s single remaining false positive is on a cherrypicking transfer protocol that uses manual `pick_up_tip()/drop_tip()` wrapping per transfer, a pattern the LLM conservatively flags as potentially unsafe even though the explicit drops force fresh tips on each operation. We interpret this as the right kind of error to keep: Layer 1 over-cautions on unusual but valid code rather than missing real violations.

The hybrid design is therefore the right architecture for assay-aware protocol validation, and it is graceful in the face of out-of-distribution assays: an LLM run without rules over-flags rather than silently failing, so for any system intended to grow with users’ own contributed rules and assay families, where the rule database is incomplete the validator over-cautions rather than under-cautions, and the failure mode of an extension is a known one.

3.2 Layer 2: PCA trajectory world model

Layer 2 builds an unsupervised world model of normal pipetting trajectories: a YOLO tip detector localises the pipette in every captured frame, and the four-keyframe tip trajectory (ASPIRATE, TRANSIT, DISPENSE, and POST-DISPENSE) of each well is projected into a low-dimensional PCA latent fit on normals only. A trajectory that reconstructs poorly against that latent is anomalous by definition; nothing in the training data tells the model what a failure looks like. We report the world-model build (tip detector and PCA), then its evaluation on a held-out trajectory set.

3.2.1 Tip detection and world-model training

YOLOv8-nano fine-tuned on 69 hand-labelled frames with tip and without reaches mAP50 of 0.995 on a held-out split after ~2 minutes on a single GPU. At inference, every captured frame yields a tight tip bounding box that is cropped, padded, and resized to a fixed 48×48 canvas before downstream scoring.

For each well, the four YOLO-cropped greyscale tip patches (ASPIRATE, TRANSIT, DISPENSE, and POST-DISPENSE) are concatenated into one 9,216-dimensional trajectory vector, and a principal-component analysis is fit on 26 normal p1000 trajectories. The variance-explained scree places the knee at $k=15$ components, retaining 97.7% of normal-trajectory variance; a k -sweep over $\{2, 5, 8, 10, 12, 15, 18, 20, 25\}$ confirms F_1 plateaus at $k=15$ and the calibrated reconstruction-error threshold collapses by twelve orders of magnitude between $k=15$ and $k=25$ as the model interpolates its small training set. Full scree and component-sweep plots are in SI Section C. Reconstruction error $\|x - \hat{x}\|_2^2$ serves as the per-well anomaly score.

3.2.2 Evaluation

We evaluate the world model on 36 planted p1000 failure trajectories, spanning three dye colours (red, yellow, water) and three failure types: partial dispense (single, accumulating, and 25/50/75% severity variants), no dispense, and air bubble. Under the leave-one-plate-out (LOPO) protocol (Section 2.3), Layer 2 reaches an average precision of 0.887 [0.799, 0.949] and an AUROC of 0.804 [0.685, 0.902] at a failure prevalence of 0.58 (Fig. 6). At the operating point, $F_1=0.708$ [0.567, 0.824], with precision 0.793 and recall 0.639: 23 of the 36 failures caught, at the cost of 6 false positives on the 10 held-out normal wells. All 95% intervals are well-level bootstrap.

The LOPO result agrees closely with the live chromophoric-cascade recall of ~0.6 (Section 3.3), so the offline number predicts what the system achieves in deployment.

The per-failure-mode breakdown (Table 1, Fig. 7) shows a clean dose-response: the more liquid a partial dispense leaves in the tip, the more reliably it is caught (25% dispensed 3/3, 50% 2/3, 75% 1/3). No dispense is caught 5/6, and air bubble is the weakest mode at 1/6: a sub-millimetre air gap gives little signal to a side view, and a reliably-visible bubble is itself hard to stage, so some injected gaps may be too small for the camera to resolve. Detection is essentially colour-independent (red and water both 8/12, yellow 7/12) because the scorer keys on the tip meniscus and trajectory geometry, not fluid colour; the geometric signal is present even for water, so the live water miss (Section 3.3) comes

from the cascade’s tight band and the VLM, not from the detector.

Table 1 Layer 2 per-failure-type recall under the leakage-free leave-one-plate-out protocol on the p1000 set. The dispense-severity gradient (25/50/75% dispensed) shows the expected dose-response as the retained-liquid column shrinks; air bubble is the weakest mode.

Failure type	Recall (LOPO)
Partial dispense, 25% dispensed (75% retained)	3/3 (100%)
Partial dispense, 50% dispensed	2/3 (67%)
Partial dispense, 75% dispensed (25% retained)	1/3 (33%)
Partial dispense, single + accumulated	11/15 (73%)
No dispense	5/6 (83%)
Air bubble	1/6 (17%)
Total	23/36 (64%)

Figure 8 shows the raw signal the detector keys on, with no class labels used at training time: normal trajectories sit at low reconstruction error and the failures spread toward higher error. The separation is real but imperfect, and that imperfection is exactly what produces the 6 false positives and 13 misses at the LOPO operating point, because a threshold fixed on one plate’s normals does not land perfectly on a new plate.

To choose a scorer, we also compared PCA against an Isolation Forest, a one-class SVM, and an uncalibrated VLM on the same p1000 set. The alternatives over-flag the normal wells whereas PCA does not, and the model-choice argument below explains why a simple linear model is the right fit at this data scale. AEGIS therefore uses linear PCA as its scorer, with a VLM second stage as the cascade’s validator (Section 2.6).

The choice of a linear model is deliberate. At $n \sim 26$ normal trajectories in 9,216 dimensions, deeper alternatives fail: autoencoders and generative world models (DreamerV3, ²² IRIS ²³) learn near-identity reconstructions that reproduce failures as faithfully as normals, while density- and boundary-based methods (Isolation Forest, one-class SVM) mis-locate the normal manifold and over-flag. Linear PCA’s low-dimensional reconstruction is the right inductive bias for a small, structured training set; train-on-normal PCA with per-run calibration is a strong baseline any guardian system should try before a deeper model.

Per-plate calibration is what makes this transfer across plates. Each plate has its own lighting, camera pose, and reagent colour, and the reconstruction-error scale itself drifts by roughly a factor of two between sessions (the calibration-well mean was ≈ 0.014 one day and ≈ 0.028 another). A single fixed threshold therefore transfers poorly and drops precision well below the LOPO value; re-anchoring it to each plate’s own first normal well recovers precision to 0.793. That calibration well is free, since any real protocol opens with standards or controls, and it assumes nothing more than that the first six wells of a run are physically normal. The live demonstration was collected in one session and so is free of cross-session drift, but any redeployment on another day or OT-2 will meet it, which is exactly what per-plate calibration absorbs.

The same protocol on the p20 channel marks the resolution limit of a front-view monitor: average precision 0.771, AUROC 0.666, and operating-point F_1 0.468 (recall 0.306), well below p1000. The p20 tip occupies far fewer pixels, so the meniscus and retained-liquid cues fall near the sensor’s floor; the sample-complexity curve is flat in training-set size (SI), confirming a reso-

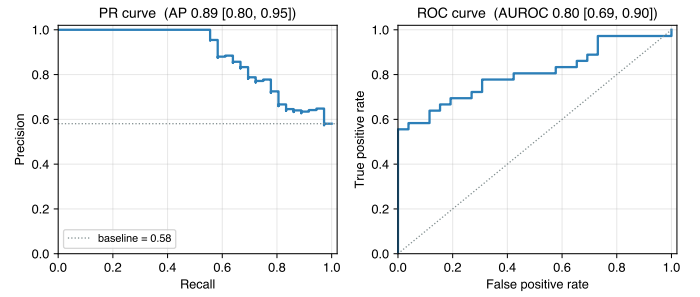


Fig. 6 Layer 2 offline detection under the leakage-free leave-one-plate-out protocol on the p1000 set. Left: precision–recall curve (average precision 0.89). Right: ROC curve (AUROC 0.80). The operating point sits at $F_1=0.71$ (precision 0.79, recall 0.64) against a failure prevalence of 0.58.

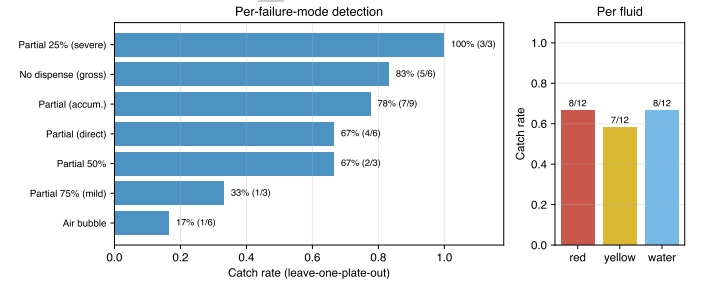


Fig. 7 Per-failure-mode and per-fluid catch rate of the trajectory PCA under leave-one-plate-out on the p1000 set. Dispense severity shows a dose-response (25%/50%/75% dispensed $\rightarrow$ 3/3, 2/3, 1/3); air bubble is the weakest mode (1/6). Catch rate is essentially colour-independent (red and water both 8/12), evidence that the scorer keys on tip geometry rather than fluid colour.

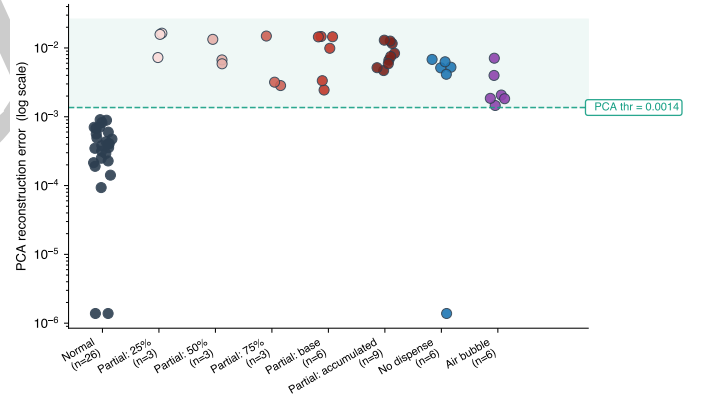


Fig. 8 Per-trajectory PCA reconstruction error on the p1000 set ($n=62$), grouped by ground-truth class: with no class labels at training time, normal trajectories concentrate at low reconstruction error while the failures (partial-dispense severities, no-dispense, air-bubble) spread toward higher error. The dashed line marks a reference threshold and the shaded band the predicted-failure region; this panel illustrates the signal the detector keys on, with the reported detection numbers coming from the operating point in Fig. 6. Two NO_TIP samples are excluded because YOLO returns no detection on blank tips; the live cascade catches those via the BLANK_FLOOR rule (Section 2.6).

lution limit rather than a data limit. This bounds where the single-camera setup is trustworthy and motivates a higher-resolution or tip-proximal camera for small-volume channels (Section 4).

3.3 Live demonstration

We exercised the cascade end-to-end on a physical OT-2 across the failure modes and three reagent colours of Section 2.6, five consecutive runs per condition, with recall reported as $X/5$ and a Wilson 95% interval. We compare two operating points: the default CASCADE (PCA triage, VLM gating only gray-zone and blank-floor wells) and ALWAYS-VLM (every post-calibration well gated by the VLM). Table 2 gives both, as failure mode $\times$ fluid $\times$ operating point; partial dispense was run on all three fluids, no-aspirate and no-tip on red only. Figure 9 shows the four-keyframe representation for a normal and a planted partial-dispense well, whose retained post-dispense column is the signal that both the PCA scorer and the VLM anchor on.

On chromophoric reagents (red and yellow) the cascade catches partial dispense in 3/5 replicates [0.23, 0.88] and always-VLM lifts this to 5/5 [0.57, 1.00] with unanimous lens agreement, on both dyes. A representative skeptical-lens diagnosis on the red-dye well visualised in Fig. 9: “Frame 4 shows a visible red column retained in the tip after dispense and lift-out, indicating incomplete dispense.”

On water the cascade fails in both directions (0/5 [0.00, 0.43]: one false pause on a normal well, the rest missed), and always-VLM reaches only 1/5 [0.04, 0.62], that catch low-confidence and mislabelled. Every water VLM verdict was medium- or low-confidence, versus high-confidence on the dyes, the signature of an underdetermined side view: a transparent fluid in a transparent tip leaves the VLM almost nothing to read. This is not a system failure but the predicted boundary of any pipette-only front view, and it motivates fusing a top-down plate view for transparent reagents.

On red dye, no-aspirate is a PCA mode, not a VLM mode: the cascade catches 3/5 [0.23, 0.88] from the transit-frame anomaly, but always-VLM falls to 2/5 [0.12, 0.77]. An empty tip *after* dispense is ambiguous, a never-filled tip and a normally-emptied one look identical from the side, so the VLM defaults to the benign read, whereas the transit frame is unambiguous. This is the live evidence for keeping PCA triage rather than routing everything to the VLM. No-tip is caught deterministically (5/5 [0.57, 1.00]) by the BLANK FLOOR rule, since YOLO returns no detection on a tipless shaft; always-VLM is marked n/a because a missing tip already routes to the VLM in either mode, so a separate run would exercise the same path.

Table 2 Live cascade catches across failure modes, reagent fluids, and operating points (five consecutive replicates per cell; recall $X/5$). CASCADE is PCA triage with VLM gating only on gray-zone or blank-floor wells; ALWAYS-VLM is every post-calibration well routed to the VLM. Partial dispense was exercised on all three fluids; no-aspirate and no-tip were exercised on red dye only. Wilson 95% intervals: 3/5 [0.23, 0.88], 5/5 [0.57, 1.00], 0/5 [0.00, 0.43], 2/5 [0.12, 0.77], 1/5 [0.04, 0.62]. One false pause occurred on a normal well in a water cascade run (Limitations).

Failure mode	Cascade			Always-VLM		
	Red	Yellow	Water	Red	Yellow	Water
Partial dispense	3/5	3/5	0/5	5/5	5/5	1/5
No aspirate	3/5	–	–	2/5	–	–
No tip	5/5	–	–	n/a	–	–

Always-VLM helps only where the cue is positive (the retained

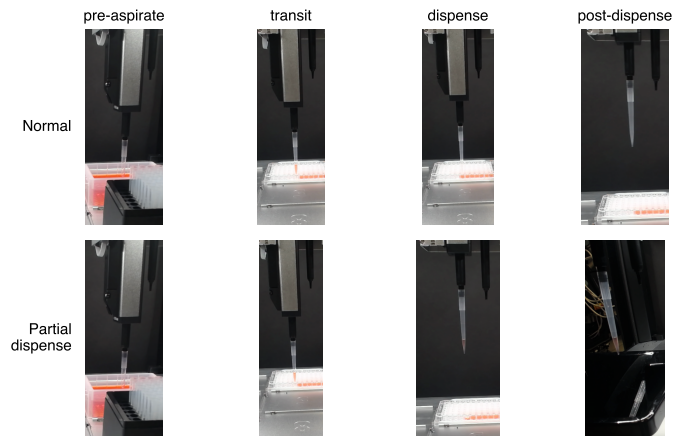


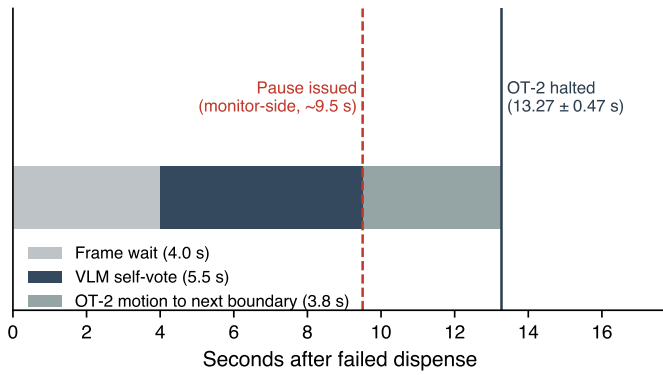
Fig. 9 Four-keyframe trajectory representation that feeds the Layer 2 cascade, captured live during a planted partial-dispense demo run on red dye. Top row: a normal well (A7). The pre-aspirate, transit, dispense, and post-dispense crops show a clean empty tip lifting away after a complete dispense. Bottom row: the planted partial-dispense well (A8) from the same run. The post-dispense crop shows the retained dye column still inside the tip after the dispense command completed, which is the dominant visual signal that the PCA scorer and the VLM gate both anchor on.

column): it lifts partial dispense to 5/5 but cannot rescue no-aspirate or water. The VLM was the source of the only false pause in the whole demo (a hallucinated failure on a normal water well) and of the no-aspirate misses (anchoring on the legitimately-empty post-dispense frame).

This is why the cascade has its shape: PCA is never allowed to pause on its own (which would over-flag unusual-but-valid wells such as empty controls), and the VLM, unreliable as a standalone scorer, only validates. Neither layer carries a precision contract it cannot meet alone. The 3-assessment self-vote prompt applies the same idea inside one call, an any-fail rule over multiple lenses beat a multi-call temperature ensemble in our tests, because correlated temperature draws suppress minority-correct catches on borderline reagents.

The cascade also exposes a cost dial. A 3-assessment self-vote VLM call costs $\approx \$0.108$ per well on Claude Opus 4.7. In the default cascade only $\sim 15.7\%$ of wells are routed to the VLM, so a 96-well plate costs $\approx \$1.63$; always-VLM routes every well at $\approx \$10.33$ per plate. The default cascade suits chromophoric reagents where PCA triages reliably; always-VLM is the safety net when it cannot (transparent reagents, low contrast). The break-even scales with reagent contrast, giving the operator an explicit dial rather than one baked-in threshold.

Latency to the OT-2 halt has two parts. The monitor-side latency (failed dispense to pause issued) is deterministic: a 4.0 s frame-capture wait plus a 5.5 s VLM call, ~ 9.5 s. The effective-halt latency (to the robot actually stopping) is set by the next protocol boundary, drop-tip in our demos, and measures 13.27 ± 0.47 s across 9 paused runs (Fig. 10). One honest limit: for protocols that share a tip across consecutive wells, that boundary can arrive before the ~ 9.5 s monitor latency, so AEGIS would pause but not prevent immediate-next-well contamination; faster triage



Effective halt is bounded by the next OT-2 protocol boundary (here: dropTip-in-trash).
In shared-tip protocols this boundary may be <5 s; AEGIS would not prevent
contamination on the immediate next well (honest limit).

Fig. 10 Latency decomposition from failed dispense to OT-2 effective halt. Monitor-side latency (failed dispense to pause API call issued) is ~ 9.5 s: a 4 s frame-flush wait plus a 5.5 s self-vote VLM call. The OT-2 then applies the pause at the next protocol boundary; effective-halt latency in the demo protocols is 13.27 ± 0.47 s ($n=9$ paused runs).

(signal-driven frame flush, a smaller vision model) is future work.

3.4 Discussion: findings and limitations

3.4.0.1 Detection is failure-mode-dependent, which is why the cascade has two stages. No single detector covers every failure. Partial dispense is a VLM strength (the retained column is a direct cue; always-VLM 5/5 on dyes). No-aspirate is the opposite, a PCA mode (cascade 3/5 vs always-VLM 2/5): an empty tip after dispense is ambiguous to the VLM, but the transit frame is not. No-tip is deterministic via BLANK_FLOOR, which fires only for a genuinely tipless shaft, since YOLO localises an empty-but-present tip (so no-aspirate does not floor). Routing everything to the VLM would lose no-aspirate; trusting PCA alone would forfeit precision, so AEGIS keeps both.

3.4.0.2 Transparent reagents fail for two reasons, only one a camera problem. The water result (cascade 0/5, always-VLM 1/5) is a principled limit with two causes that need different fixes. One is geometric: a transparent fluid in a transparent tip, seen from the side, gives almost no contrast, and no sensor fixes that. The other is resolution: the consumer webcam puts too few pixels on the tip, the same bottleneck that caps p20; a higher-resolution or tip-proximal camera helps here and on p20/p300, but not the geometry. The robust fix is a top-down post-run plate view (a prospective Layer 3) that measures dispensed volume independently of tip transparency. The VLM's uniformly low-confidence water verdicts are it reporting this uncertainty honestly rather than guessing.

3.4.0.3 Reporting choices at small N . We report live recall as $X/5$ with a Wilson interval; a Wald error is inappropriate here, collapsing to zero width at 0/5 and 5/5. The confidence-interval-backed claim remains the offline LOPO number (AP 0.887, F_1 0.708), with the live matrix the demonstration it predicts. False pauses are reported as raw counts, not a rate, which would be uninformatively wide at $N=5$. Per honest- N reporting, the

five replicates were consecutive runs with a fixed detector (Section 2.6) and no run was excluded on its outcome; an unrelated calibration-well false pause was resumed so the planted failure still ran.

3.4.0.4 The cascade improvement is multi-factor. We caution against over-attributing the gap between the initial single-replicate cascade and the calibrated design here: the redesign changed three things at once, the calibration count ($n=3 \rightarrow 6$), the silent-pass band (raw $[\min, \max] \rightarrow$ robust median $\pm 0.7 \cdot \text{MAD}$), and the fail-well position, so the gain reads as their joint effect, not any one alone. Isolating them is a small future ablation.

4 Conclusions

We have presented AEGIS, a two-layer guardian architecture for the Opentrons OT-2 and similar open-source liquid handlers that addresses the silent-failure problem in self-driving laboratories: failures that occur in either the protocol code or the physical execution and that routinely escape detection until a downstream assay fails or an operator notices an empty well. Layer 1 reads the protocol's Python source before each run and flags assay-level violations against a curated rule database; Layer 2 watches the pipette through an off-the-shelf camera at runtime, scoring per-well trajectories with a small PCA world model and routing uncertain wells to a vision-language model gate that arbitrates pause decisions.

Contributions

Our primary contribution is the two-layer guardian architecture above, which retrofits onto the existing OT-2 stack with off-the-shelf cameras and no robot-side modifications. The second contribution is the cascade design pattern at Layer 2 itself: PCA-as-triage with per-run calibration, VLM-as-validator, and neither layer permitted to halt a run on its own signal, so that each layer's known failure mode is gated by the other and the operator has an explicit cost-coverage dial between the default and always-VLM operating points. We demonstrate the cascade end-to-end on a physical OT-2 with effective-halt latency 13.27 ± 0.47 s ($n=9$ paused runs), catching planted partial-dispense failures across three reagent contrasts (red, yellow, water) and planted no-aspirate and no-tip failures on red dye, the latter two being independent of reagent identity (the failure signatures are mechanical, not optical). To support reproducibility and community extension, we release under the MIT license a 24-protocol benchmark with planted assay-level bugs (to our knowledge the first public corpus of liquid-handling failures with assay-level rule annotations), a 22-rule assay rule database, the YOLOv8-nano tip-detector weights, the trajectory dataset, the live demo harness, and reproduction scripts for every table and figure in this paper; a separate negative-results record (SI Section B) documents alternatives that did not work, recorded so the next group does not retread the same paths.

Key findings and insights

At Layer 1, all four LLM backends achieve 100% bug recall on the single-bug benchmark, so the LLM is never the detection bottle-

neck; the rule database carries the precision contract, and an LLM run without rules over-flags rather than silently failing, which makes the system graceful in the face of out-of-distribution assays. At Layer 2, linear PCA with per-run calibration outperforms both deep generative world models (which overfit on 26 normal trajectories and reconstruct anomalies as faithfully as normals) and boundary or density baselines (Isolation Forest, one-class SVM), a transferable lesson for any guardian trained on a small normal corpus. Per-run median $\pm 0.7 \cdot \text{MAD}$ anchoring further absorbs cross-session drift in the reconstruction-error scale, which we observed to shift by approximately a factor of two between two demonstration sessions on the same hardware; the cascade absorbed this drift without retraining.

Neither layer is trustworthy as a standalone scorer. The PCA can silently pass unusual-but-valid wells; the VLM hallucinates failures on low-contrast crops and misses real failures by anchoring on the wrong frame. The cascade gates each layer's known failure mode through the other and exposes an explicit cost-coverage dial: the default cascade routes only $\sim 15.7\%$ of post-calibration wells to the VLM and costs an estimated \$1.63 per 96-well plate, while the always-VLM operating point routes every well and costs \$10.33 per plate. The two-layer complementarity is architectural by construction: Layer 1 sees the protocol code, Layer 2 sees the physical execution, and neither layer can substitute for the other. Finally, front-view monitoring of transparent reagents is a principled physical limit; AEGIS surfaces this in the VLM's own low-confidence reasoning rather than hallucinating a verdict.

Current limitations

Five limitations bound the present work. First, the primary evaluation is on the p1000 single-channel pipette; we additionally quantify the p20 channel under the same leave-one-plate-out protocol and find it near the resolution limit of crop-based PCA at our camera setup (F_1 0.47 versus 0.71 on p1000, with a flat sample-complexity curve indicating a resolution rather than a data limit), while the p300 and the 8-channel multichannel head are likely tractable with the same pipeline but as yet untested (8-channel runs are in progress). Second, the $n=6$ per-run calibration with a median $\pm 0.7 \cdot \text{MAD}$ band is robust to a single anomalous calibration well but not to a cluster of mis-behaving calibration wells; cross-run normal priors that accumulate as more sessions log their data would shrink this variance further. Third, front-view monitoring of transparent reagents is a principled limit on volume-based failures: a transparent fluid in a transparent tip is nearly invisible from the side. A higher-resolution or tip-proximal camera would help the resolution component of this limit but not the underlying geometry, so a top-down post-run plate view (Layer 3) is the robust fix. Fourth, the monitor-side latency of ~ 9.5 s can exceed the next protocol-level boundary in protocols that share a tip across consecutive destination wells; AEGIS still halts the run but would not prevent immediate-next-well contamination in those cases. Fifth, the live demonstration is on a single OT-2 across three planted-failure modes and three reagent contrasts; air-bubble injection proved difficult to stage reliably in a live run

and so could not be tested live (it is evaluated only in the offline set, where it is the weakest mode), and coverage of new assays and multi-protocol scheduling on a heterogeneous fleet remain future work.

Future work

The most important future direction is strengthening the live cascade so that failures are caught the moment they occur, reducing both the post-run audit burden and the wasted reagent and time that result when a failure is only detected after a long run completes. As more normal trajectories accumulate across deployments, a stronger world model (a fine-tuned successor to linear PCA, or an online streaming PCA that adapts continuously and removes the explicit calibration phase) will absorb more of the routing decision, the gray-zone fraction will shrink, and the per-plate VLM call count will drop without changing the cascade's safety contract. A tiered VLM gate where a smaller and faster vision model triages the gray-zone and the heavier model arbitrates only the ambiguous remainder is feasible today and cuts the current ~ 9.5 s monitor-side floor by an estimated $3\text{--}5\times$ on the routine majority of paused wells. On the sensing side, an active camera that zooms or pans to track the tip in real time, steered by the existing YOLO tip detector, would place far more pixels on the tip and directly attack the resolution limit behind both the p20 small-pipette and transparent-reagent cases.

More broadly, the linear PCA scorer is a deliberate baseline, not a ceiling: it is the right model at $n \sim 26$ normal trajectories, where deeper alternatives overfit, but it is the component most likely to improve as data grows. With more normal trajectories and, crucially, more labelled failures, a fine-tuned autoencoder or deep world model, or a foundation or vision-language model fine-tuned on this corpus, should raise detection on exactly the modes PCA handles least well today (air bubble, transparent reagents, small pipettes), while the train-on-normal, per-plate-calibrated cascade around it stays the same. AEGIS is thus a data-efficient starting point whose accuracy scales with the datasets its own deployment generates.

In the meantime, a post-run audit layer (L3) is a near-term complement that re-scores every well after the run with a slower-and-stricter ensemble and surfaces silent-passes for human review. A prototype already caught the water partial-dispense case retrospectively, and the role of the audit naturally diminishes as the live cascade strengthens.

The longest-term extension is closing the loop from detection to autonomous correction, with two distinct cases. Some failures are recoverable at the protocol level: re-aspirating from a fresh reservoir, swapping tips from a remaining tip rack, or retrying a transfer. For these, an AEGIS agent with sufficient context could trigger the correction itself or hand control to a protocol-rewriting predictive model that proposes a corrective sub-protocol on the fly. Hardware-level failures (a missing tip on the rack, a clogged tip mid-run, deck contamination, a knocked-over reservoir) are outside the OT-2's own actuation envelope, so true autonomy in those cases would require a physical-intervention layer such as a humanoid arm or robotic operator that loads tips, replaces con-

sumables, and clears mechanical faults. AEGIS as the perception layer fits naturally into either trajectory: the same trajectory features and VLM verdicts that pause the run today could route to a corrective sub-protocol for the software-recoverable class, or to a physical-intervention request for the hardware-level class.

Two narrower directions follow from the Layer 1 and benchmark sides. The Layer 1 validator is currently a single LLM call per protocol; a multi-call ensemble is the obvious lever for further reducing false positives at higher cost, though we did not pursue it here because the Layer 2 VLM-gate analogue (multi-call temperature ensemble) produced correlated errors that majority-vote suppressed on minority-correct catches in our small evaluation. The released benchmark invites community contributions in both data and rules; fine-tuning smaller open-source models on this corpus would lower per-protocol API cost and bring both layers into reach for fully-local deployment.

Broader impact

The two failure surfaces AEGIS targets occur in every open-source self-driving laboratory we are aware of, and they routinely escape detection until a downstream assay fails or an operator notices an empty well. Defense in depth across non-overlapping surfaces, namely code-level validation and physical-execution monitoring, reduces the silent-failure rate that otherwise compounds across runs in an autonomous lab. The cascade pattern itself, cheap triage with an expensive validator and neither layer acting alone, generalises beyond AEGIS to any deployment where a strong but imperfect cheap model and a more capable but fallible LLM gate are both available. By releasing the benchmark, the rules, the YOLO weights, the trajectory dataset, and the live demo harness under a permissive license, we lower the floor for adding a guardian layer to any community OT-2 setup; there is little reason for a new open-source self-driving laboratory to operate fully open-loop.

Author contributions

Conflicts of interest

There are no conflicts to declare.

Data availability

The AEGIS source code, protocol benchmark (24 OT-2 protocols across 5 assay families), assay rule database, Layer 2 trajectory dataset (128 labelled samples across 3 reagent contrasts and 3 physical failure types), trained YOLOv8 weights, and reproduction scripts are available at [GitHubURLtobeaddedonrelease] under the MIT license. Raw camera captures are deposited at [Zenodo DOI to be assigned upon acceptance].

Acknowledgements

We thank the Opentrons community for protocol examples used in our correct-protocol benchmark, and the Academy framework team at Globus Labs for early access and integration support. This work was supported by [funding sources to be added]. Argonne National Laboratory's work was supported by the U.S. Department of Energy, Office of Science, under contract DE-AC02-

06CH11357.

Notes and references

- 1 G. Tom, S. P. Schmid, S. G. Baird, Y. Cao, K. Darvish, H. Hao, S. Lo, S. Pablo-García, E. M. Rajaonson, M. Skreta, N. Yoshikawa, S. Corapi, G. D. Akkoc, F. Strieth-Kalthoff, M. Seifrid and A. Aspuru-Guzik, *Chemical Reviews*, 2024, **124**, 9633–9732.
- 2 N. J. Szymanski, B. Rendy, Y. Fei, R. E. Kumar, T. He, D. Milsted, M. J. McDermott, M. Gallant, E. D. Cubuk, A. Merchant, H. Kim, A. Jain, C. J. Bartel, K. Persson, Y. Zeng and G. Ceder, *Nature*, 2023, **624**, 86–91.
- 3 D. A. Boiko, R. MacKnight, B. Kline and G. Gomes, *Nature*, 2023, **624**, 570–578.
- 4 S. Kwok and R. Higuchi, *Nature*, 1989, **339**, 237–238.
- 5 M. M. Bradford, *Analytical Biochemistry*, 1976, **72**, 248–254.
- 6 J.-H. Zhang, T. D. Y. Chung and K. R. Oldenburg, *Journal of Biomolecular Screening*, 1999, **4**, 67–73.
- 7 Y. Gao, Y. Luo, W. Li, Y. Lan, H. Jiang, Y. Chen, X. Yi, L. Fu, M. Yang and T. Si, *bioRxiv*, 2025.
- 8 Opentrons Labworks, *Opentrons AI Protocol Designer and Protocol Library*, <https://opentrons.com/ai>, 2024, Accessed 2026-05-20.
- 9 A. M. Bran, S. Cox, O. Schilter, C. Baldassari, A. D. White and P. Schwaller, *Nature Machine Intelligence*, 2024, **6**, 525–535.
- 10 K. Darvish, M. Skreta, Y. Zhao, N. Yoshikawa, S. Som, M. Bogdanovic, Y. Cao, H. Hao, H. Xu, A. Aspuru-Guzik, A. Garg and F. Shkurti, *Matter*, 2025.
- 11 M. Skreta, N. Yoshikawa, S. Arellano-Rubach, Z. Ji, L. B. Kristensen, K. Darvish, A. Aspuru-Guzik, F. Shkurti and A. Garg, *arXiv preprint arXiv:2303.14100*, 2023.
- 12 O. O'Donoghue, A. Shtedritski, J. Ginger, R. Abboud, A. Ghareeb, J. Booth and S. G. Rodrigues, *arXiv preprint arXiv:2310.10632*, 2023.
- 13 J. O. Curtis, D. Ricketts, S. Mansoor, C. Cremer and P. Brisk, *Proceedings of the ACM on Programming Languages (OOPSLA)*, 2018, pp. 1–31.
- 14 B. Hsu, P. Setty, R. Butler, R. Lewis, C. Stone, R. Weinberg, T. Brettin, R. Stevens, I. Foster and A. Ramanathan, *Digital Discovery*, 2026.
- 15 Hamilton Company, *Total Aspiration and Dispense Monitoring (TADM) Technology*, <https://www.hamiltoncompany.com/automated-liquid-handling/platform/microlab-star-products/tadm>, 2024, Accessed 2026-05-20.
- 16 Tecan Group, *Process Monitored Pipetting (PMP) and Liquid-Level Detection on Tecan Fluent*, <https://www.tecan.com/fluent-laboratory-automation>, 2024, Accessed 2026-05-20.
- 17 S. U. Khan, V. K. Müller, R. J. N. Frandsen and M. Mansourvar, *Applied Intelligence*, 2025.
- 18 Y. Yin, J. Lei and W. Tao, *Electronics*, 2024, **13**, 2836.
- 19 Z. Zhou, S. Veeramani, F. Munguia-Galeano, H. Fakhrudeen and A. I. Cooper, *Communications Chemistry*, 2025, **8**, 384.

Table 3 Academy fleet-scaling sweep on the 16-protocol benchmark, simulated OT-2 backends. Detection is L1+L2 verdict agreement on each protocol.

N robots	Wall (s)	Speedup	% ideal	Detection
1	157.5	1.00×	100%	15/16
2	86.0	1.83×	92%	15/16
4	45.4	3.47×	87%	15/16
8	29.2	5.39×	67%	15/16
16	20.1	7.84×	49%	15/16

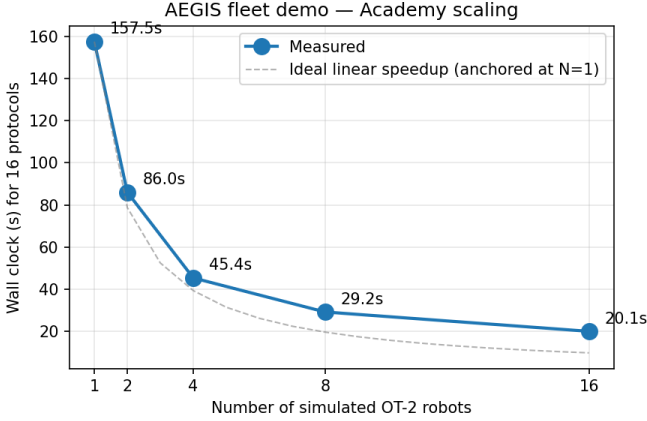


Fig. 11 Academy fleet scaling on the 16-protocol benchmark using simulated OT-2 backends. Wall-clock time drops 7.84× from $N=1$ to $N=16$ agents; detection is invariant at 15/16 across the sweep. Sublinearity at high N is floored by per-protocol Layer 1 LLM latency.

- 20 F. Domberg and G. Schilbach, *arXiv preprint arXiv:2503.02552*, 2025.
- 21 M. Ho, M. F. Ginting, I. R. Ward, A. Reinke, M. J. Kochenderfer, A.-a. Agha-Mohammadi and S. Omidshafiei, *arXiv preprint arXiv:2602.16182*, 2026.
- 22 D. Hafner, J. Pasukonis, J. Ba and T. Lillicrap, *arXiv preprint arXiv:2301.04104*, 2023.
- 23 V. Micheli, E. Alonso and F. Fleuret, *International Conference on Learning Representations (ICLR)*, 2023.
- 24 N. Hansen, H. Su and X. Wang, *International Conference on Learning Representations (ICLR)*, 2024.
- 25 K. Darvish, A. Sohal, A. Mandal, H. Fakhruddin, N. Radulov, Z. Zhou, S. Veeramani, J. Choi, S. Han, B. Zhang, J. Chae, A. Wright, Y. Wang, H. Darvish, Y. Zhao, G. Tom, H. Hao, M. Bogdanovic, G. Pizzuto, A. I. Cooper, A. Aspuru-Guzik, F. Shkurti and A. Garg, *arXiv preprint arXiv:2601.13232*, 2026.
- 26 P. Bergmann, M. Fauser, D. Sattlegger and C. Steger, *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2019, pp. 9592–9600.
- 27 K. Roth, L. Pemula, J. Zepeda, B. Schölkopf, T. Brox and P. Gehler, *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2022, pp. 14318–14328.
- 28 Z. Zhang, C. Yue, H. Xu *et al.*, *arXiv preprint arXiv:2509.08820*, 2025.
- 29 S. Bai, W. Song, J. Chen *et al.*, *arXiv preprint arXiv:2512.22983*, 2025.

30 J. G. Pauloski *et al.*, *arXiv preprint*, 2025.

A Supplementary Information

A.1 A. Multi-robot fleet deployment via Academy

A.1.0.1 Architecture. To support multi-robot deployment, AEGIS wraps both layers as a single AEGISAgent class (`aegis/agents/aegis_agent.py`) exposing one `@action process_protocol(...)` entry point. The agent owns one `ProtocolValidator` (Layer 1) and one `Layer2Inference` (Layer 2) per robot. A fleet of N such agents is launched under `academy.manager.Manager`³⁰ with the `LocalExchangeFactory` and a `ThreadPoolExecutor`; protocols are partitioned across agents and processed in parallel. No code in either layer is aware of the fleet: scaling AEGIS is launching more agents.

A.1.0.2 Scaling sweep (simulated fleet). We swept $N \in \{1, 2, 4, 8, 16\}$ AEGISAgent instances on a 16-protocol benchmark workload using simulated OT-2 backends. Detection is invariant at 15/16 across the sweep; wall-clock time drops 7.84× from $N=1$ to $N=16$ (Table 3, Fig. 11). Sublinearity at high N is gated by per-protocol Layer 1 latency (single LLM call, $\sim 8\text{--}10\text{s}$ on Opus): once each agent has only one protocol to process, the slowest single LLM call floors the wall clock. The architectural implication is that scaling AEGIS to a real heterogeneous fleet is a deployment exercise rather than a redesign; the per-robot agent already encapsulates the necessary state. The scaling sweep is on simulated robots and does not characterise heterogeneous-hardware behaviour, which we list as a future-work item in the main text.

A.2 B. Negative results: alternatives that did not work

In the course of building Layer 2 we evaluated several alternatives that did not work and are documented here for completeness:

- *Multi-call VLM ensemble with temperature sampling.* Three independent VLM calls with $T > 0$ and majority vote over their verdicts. The three draws produced highly correlated errors on borderline cases; majority vote then suppressed minority-correct catches. The in-prompt 3-assessment self-vote that we use in the live cascade resolves this by aggregating with an ANY-FAIL rule across distinct lenses inside a single call.
- *Post-dispense single-frame PCA.* Scoring only the post-dispense frame (rather than the four-frame trajectory) cannot detect failures in which the tip is empty in both normal and failure conditions (no aspirate, no dispense, air bubble during transit). The four-frame trajectory representation supplies the temporal context that the single-frame view lacks.
- *Convolutional autoencoder world model.* Even with aggressive bottlenecks, autoencoders trained on 26 normal trajectories learn near-identity reconstructions and reconstruct failure trajectories as faithfully as normals, vanishing the discriminator. Linear PCA’s lower-capacity inductive bias is the right match for the small-sample regime.

- *Crop-based PCA on the p20 pipette.* The 20 μ L pipette tip is geometrically thin at our camera setup, so cropped trajectory vectors of failure and normal wells are only weakly separable: the same leave-one-plate-out protocol yields average precision 0.77, AUROC 0.67, and operating-point F_1 0.47 (recall 0.31), well below the p1000 numbers (0.89 / 0.80 / 0.71). The sample-complexity curve is flat in the number of training normals, indicating a resolution rather than a data limit. A higher-resolution front view or a different representation (e.g., a VLM-only path, or a specialised meniscus detector) is the natural next step.
- *Top-down iPhone capture for in-flight monitoring.* The OT-2 gantry occludes the top-down plate view at most pipetting moments; even after homing, the gantry’s resting position can block portions of the plate. We retain the top-down capture only for post-run plate state and document the in-flight occlusion as a deck-geometry limitation.
- *PCA-projected Mahalanobis (15-dim).* Computing Mahalanobis distance in the 15-dimensional PCA-projected space (rather than the 32-dimensional hand-crafted feature space) fails because failures project to lower-energy regions of the PCA latent than normals; the very directions PCA selects are the directions in which failures concentrate close to the origin, inverting the distance metric.

A.3 C. PCA component selection for the trajectory world model

The PCA component count k for the trajectory world model is chosen at the variance-explained knee, where F_1 on the held-out set plateaus and the calibrated reconstruction-error threshold has not yet collapsed into floating-point noise. The variance-explained scree (Fig. 12) places the knee at $k=15$, retaining 97.7% of normal-trajectory variance. A k -sweep over $\{2, 5, 8, 10, 12, 15, 18, 20, 25\}$ confirms F_1 rises monotonically up to $k=15$ and plateaus thereafter, while the calibrated threshold collapses by roughly twelve orders of magnitude between $k=15$ and

$k=25$ as the model begins to interpolate its 26-trajectory training set (Fig. 13). Beyond the knee the detector degenerates: reconstruction error of both normals and failures shrinks to floating-point noise and the discriminator vanishes.

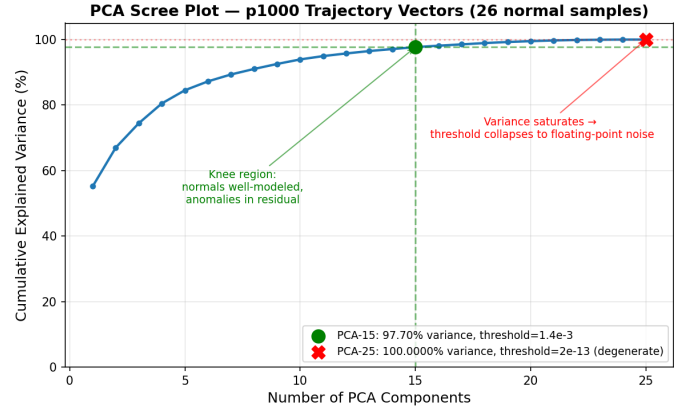


Fig. 12 Variance-explained scree for the trajectory PCA. The knee at $k=15$ retains 97.7% of normal variance; beyond $k=20$ the model interpolates its 26-trajectory training set and the calibrated reconstruction-error threshold collapses to floating-point noise.

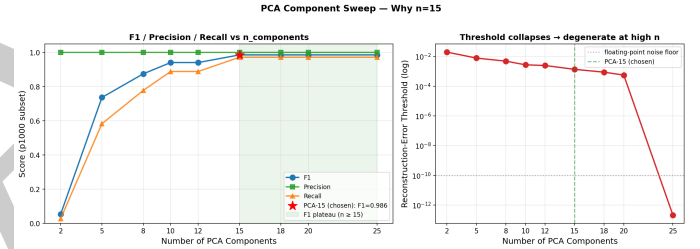


Fig. 13 PCA component sweep. F_1 , precision, and recall on the p1000 set across k , alongside the calibrated threshold. F_1 plateaus at $k=15$; the threshold collapses at high k as the model interpolates the training set.