MICROBIO-BENCH: CAN VISION LANGUAGE ACTION MODELS WORK IN MICROBIOLOGY LABS?

Anonymous authors
Paper under double-blind review

ABSTRACT

Vision–language–action (VLA) models offer a way to automate laboratory procedures that require precise manipulation and decisions about when a tool can still be used. In microbiology, a disposable tool whose working end touches a prohibited surface must be discarded before further sample contact, even if it remains correctly positioned for use. We introduce MicroBio-Bench, a simulation environment and benchmark for evaluating how VLA models compose laboratory operations and respond when contamination changes the required action. The suite connects 60 tasks across atomic actions, local operations, and multi-operation tasks, and tracks tool contacts to determine when disposal is required. A contamination-response study tests whether policies learn disposal, retain normal tool use, and generalize to new contact sites. We evaluate three models over 13,050 simulation episodes, complemented by hardware tests. Atomic-action success averages 89.0–97.3%, yet mean multi-operation task success peaks at 24.8%. Response demonstrations yield 100% disposal success for loops and 86–92% for spreaders at seen sites, while unseen-site success falls to 0–20% and 0–18%, respectively. These findings identify workflow composition and generalization of contamination response as key challenges for VLA models in microbiology labs.

1 INTRODUCTION

Robots are beginning to automate scientific experiments (Burger et al., 2020; Boiko et al., 2023), and vision–language–action (VLA) models offer a way to learn laboratory operations from demonstrations (Brohan et al., 2023; Kim et al., 2025; Octo Model Team et al., 2024). Applying these models to microbiology involves carrying out sequences of dependent operations. A successful grasp, for example, may leave a vessel misaligned for subsequent insertion (Ross et al., 2011). Completing a procedure therefore requires each operation to leave a suitable starting state for the next.

In microbiology, this dependence on earlier actions also extends to a tool's contact history. A disposable tool may be correctly positioned above a sample yet need to be discarded because its working end has touched the workbench. Continued use could transfer contaminants to the sample, which aseptic technique seeks to prevent (Sanders, 2012). The next action thus depends on both the physical state left by previous operations and whether earlier contact permits further tool use. These dependencies motivate two questions: **can VLA models compose precise operations into reliable procedures, and can they change their actions when contamination makes further tool use invalid?**

Evaluating these questions requires comparing policy behavior across isolated operations and longer procedures, as well as during normal execution and after prohibited contact. Repeated physical trials require careful setup and consume disposable tools and culture media. Simulation supports these comparisons by making starting states and contact conditions repeatable. Existing laboratory benchmarks evaluate instrument interaction and workflow completion (Lan et al., 2025; Li et al., 2025; Tang et al., 2026). For microbiology, this evaluation also needs to account for contacts that change whether a tool can still be used.

We introduce MicroBio-Bench, a simulation environment and benchmark that evaluates these two requirements in shared microbiology workflows (Figure 1). Nested tasks connect atomic actions, local operations, and multi-operation procedures. The same policy can be tested on an operation in

Reviewers: please read the Reviewer Guidelines (iclr.cc/Conferences/2027/ReviewerGuidelines) and the AI Policy for Reviewers (iclr.cc/Conferences/2027/AIPolicyForReviewers). If you used AI to expand, edit, or polish your review, please provide the input text to the LLM. Better still, consider skipping the LLM and submitting your original text: we, and the authors, are much more interested in your unedited thoughts than in what an LLM has to say. AI-assisted or not, you are putting your name and reputation behind your review: LLM-generated falsehoods, hallucinations or misrepresentations are subject to disciplinary action, which may include desk-rejecting all papers you have authored.

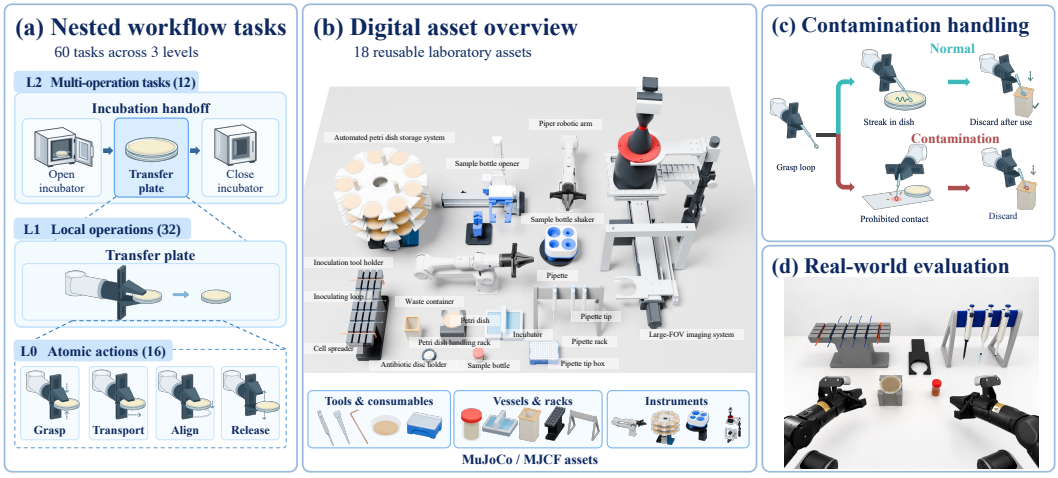


Figure 1: **MicroBio-Bench.** (a) Shared workflows connect 16 atomic actions (L0), 32 local operations (L1), and 12 multi-operation tasks (L2). (b) Eighteen expert-built assets provide tools, vessels, and instruments. (c) Aseptic constraints change the required action from inoculation to disposal after prohibited working-end contact. (d) A dual-arm laboratory setup supports real-world evaluation of microbiology manipulation.

isolation and within its enclosing procedure, where its preceding actions determine the operation's starting state.

The benchmark also evaluates how models respond to tool contamination. Standard demonstrations include disposal at the end of a normal workflow; contamination-response demonstrations place disposal after prohibited contact. Comparing policies trained with and without these response demonstrations tests whether they learn when disposal is required. Evaluation on normal workflows and held-out contact sites then examines whether they preserve ordinary tool use and extend the response to new contacts.

Our contributions are:

- We introduce MicroBio-Bench, a reusable simulation environment and benchmark for microbiology manipulation, with 18 expert-built MJCF assets, 60 nested tasks (16 atomic actions, 32 local operations, and 12 multi-operation tasks), and an automated scene-construction pipeline.
- We model contamination in single-use inoculating loops and spreaders: after prohibited working-end contact, the tool must be discarded before further sample or agar contact. Our five-group study shows that policies trained with response demonstrations achieve high disposal success at seen contact sites, but generalize poorly to held-out sites and may impair normal workflow performance.
- We evaluate $\pi_{0.5}$, RDT, and π_0 over 13,050 simulated episodes and four hardware operations. Mean success rates range from 89.0% to 97.3% on atomic tasks but reach at most 24.8% on multi-operation tasks. Cross-level tests further show that the same policy can perform differently on an operation when it is tested alone versus executed within a longer task.

2 Related Work

Laboratory simulation. Chemistry3D combines robotic interaction with chemical-process simulation, AutoBio models biological instruments, and Pipette provides editable wet-lab assets (Li et al., 2024; Lan et al., 2025; Liu et al., 2026). LabUtopia generates procedural layouts, while Labimus assembles scenes from experimental procedures (Li et al., 2025; Wu et al., 2026). RoboGen combines task-conditioned scene generation with physical checks (Wang et al., 2023). These laboratory benchmarks evaluate manipulation and workflow execution but leave aseptic tool use untested (Table 1). MicroBio-Bench adds contact-dependent tool eligibility to reusable microbiology scenes, enabling evaluation of the response required after contamination.

Table 1: **Comparison with laboratory benchmarks.** ✓: included in the reported benchmark; ✗: not reported. Real world denotes robot-policy hardware evaluation.

Benchmark	Domain	#Tasks	Real world	Dual arm	Nested tasks	Auto scenes	Contam. test
Chemistry3D	Chemistry	5	✗	✗	✗	✗	✗
AutoBio	Biology	16	✗	✓	✗	✗	✗
LabUtopia	Scientific lab	30	✗	✗	✓	✓	✗
LabDex	Chemistry	40	✓	✗	✓	✗	✗
Labimus	Chemistry	7	✗	✓	✓	✓	✗
Pipette	Wet lab	12	✗	✗	✗	✓	✗
MicroBio-Bench	**Microbiology**	**60**	✓	✓	✓	✓	✓

Compositional robot evaluation. CALVIN evaluates long-horizon language-conditioned manipulation, and LIBERO studies knowledge transfer across manipulation tasks (Mees et al., 2022; Liu et al., 2023a). GTI composes demonstrated behaviors through shared regions of state space (Mandlekar et al., 2020), while offline imitation-learning studies examine how demonstration quality and policy design affect multi-stage manipulation (Mandlekar et al., 2022). Laboratory benchmarks extend evaluation through task concatenation, hierarchical levels, and intermediate progress measures (Lan et al., 2025; Li et al., 2025; Tang et al., 2026; Wu et al., 2026). REFLECT uses robot-experience summaries to explain failures and guide correction (Liu et al., 2023b). MicroBio-Bench pairs isolated operations with their enclosing multi-operation tasks, evaluates fixed checkpoints across levels, and quantifies failure patterns in selected workflows.

Contact-dependent procedures. Aseptic technique governs contact between tools, samples, and culture media (Sanders, 2012). SafeManip specifies temporal safety properties for household manipulation, including preventing contact with clean surfaces after contamination until sanitization, and monitors existing VLA policies for violations (Huang et al., 2026). MicroBio-Bench studies how policies learn the required response in microbiology: prohibited working-end contact makes a single-use tool ineligible and changes the next action from sample manipulation to immediate disposal. Standard demonstrations already include disposal at the end of the workflow; response demonstrations teach disposal immediately after prohibited contact. Our five-group protocol measures response learning, retention of normal execution, and generalization to held-out contact sites.

3 MICROBIO-BENCH

MicroBio-Bench supports repeatable training and evaluation of precise microbiology manipulation. Workflows span sample preparation, plate handling, streaking or pipetting and spreading, disposal, incubation handoff, local imaging, and disc placement. Nested tasks evaluate operation composition; a contamination-response study uses selected loop and spreader workflows to test disposal after prohibited contact.

3.1 EVALUATING NESTED WORKFLOWS

Each task specifies a starting state, instruction, completion conditions, and required steps. **L0** contains 16 atomic-action instances: grasping, transport, alignment, or release. **L1** contains 32 local operations composed of primitives. **L2** contains 12 multi-operation tasks chaining local operations in continuous rollouts. Incubation handoff combines incubator opening, plate transfer, and closure; plate transfer combines grasping, transport, alignment, and release (Figure 1a).

The nesting map links shared operations (Appendix A), with L1 sequences illustrated in Appendix A.2. Lower-level tasks start with prerequisites satisfied: release begins with an object held, whereas transfer requires acquisition and positioning. Each model uses one multi-task checkpoint per level, trained on that level's pooled demonstrations.

Table 2: **Five-group contamination-response design.** I_{std}: the workflow-specific normal sequence. I_{aseptic}: the same instruction plus the conditional disposal rule. G2 uses the same contact sites as G4. Seen and held-out refer to sites included in or excluded from the contamination-response demonstrations used for Response-augmented training, respectively.

Group	Training	Test condition	Instruction	Evaluation purpose
1	Standard-only	Normal	I_{std}	Standard-workflow baseline
2	Standard-only	G4 contact sites	I_{aseptic}	Disposal baseline
3	Response-augmented	Normal	I_{std}	Normal-workflow retention
4	Response-augmented	Seen contact	I_{aseptic}	Learned disposal
5	Response-augmented	Held-out contact	I_{aseptic}	Contact-site generalization

In L2, each operation starts from the grasp and placement produced by the preceding operation, exposing the transition from prepared to policy-generated states. Shared tasks retain separate IDs because single-arm streaking and dual-arm spreading have different output action dimensions.

Using the L1 completion predicates and thresholds in Appendix A.1, we measure terminal success and partial execution. Let G_i indicate terminal completion in episode i, and k_i count consecutive scoring subtasks completed from the start of an L2 task with K ordered milestones. Success rate (SR) and ordered progress (PG) are

$$\text{SR} = \frac{100}{N}\sum_{i=1}^{N} G_i, \qquad \text{PG} = \frac{100}{N}\sum_{i=1}^{N} \frac{k_i}{K}. \tag{1}$$

SR captures completion; PG captures progress through the ordered L1 operations in the nesting map. For example, in L2-02, the milestones are plate transfer and lid removal; in L2-04, the milestones are opening the incubator, transferring the plate, and closing it.

Ordered scoring preserves procedural dependencies: completing the first two of four milestones gives 50% progress, unchanged by reaching the fourth while the third remains unmet. PG distinguishes early failures from those occurring later in the procedure.

3.2 Representing contamination and tool eligibility

We represent aseptic constraints through contact-dependent tool eligibility. Prohibited contact contaminates the working end and requires disposal, even if its appearance remains unchanged. For each critical working surface r, $q_t(r) = 1$ means eligible for use and $q_t(r) = 0$ means contaminated. Starting from $q_0(r) = 1$, we update this state as

$$q_{t+1}(r) = q_t(r)\,[1 - c_t(r)], \tag{2}$$

where $c_t(r) = 1$ denotes contaminating contact at time t, and 0 otherwise. Contamination is irreversible within an episode: the tool must be discarded before further sample or agar contact.

Surface annotations distinguish the working end from the shaft. Sample contact during loop dipping and agar contact during streaking or spreading leave $c_t(r) = 0$; working-end contact with the workbench or plate-support exterior sets $c_t(r) = 1$. In the response study, a script controls the arm to inject prohibited contact. Training demonstrations record correct post-injection disposal; at test time, the script returns control to the model to predict its next action from images and instructions. This evaluates responses to externally imposed contamination.

3.3 Evaluating contamination response

The contamination-response study uses the loop and spreader workflows (Sanders, 2012). For feasible evaluation with a shared disposal endpoint, the loop sequence uses L1-06/07/08 from L2-03: pickup and sample dipping, streaking, and disposal, omitting plate closure (L1-09). The spreader sequence uses L1-25/26/27 from L2-10: pickup and transfer to agar, spreading, and disposal. In response demonstrations, the tool is discarded after contamination.

For each backbone, response-study policies train on one tool-specific sequence, unlike the main L2 checkpoint, which trains jointly on all 12 L2 tasks. We fine-tune **Standard-only** on standard

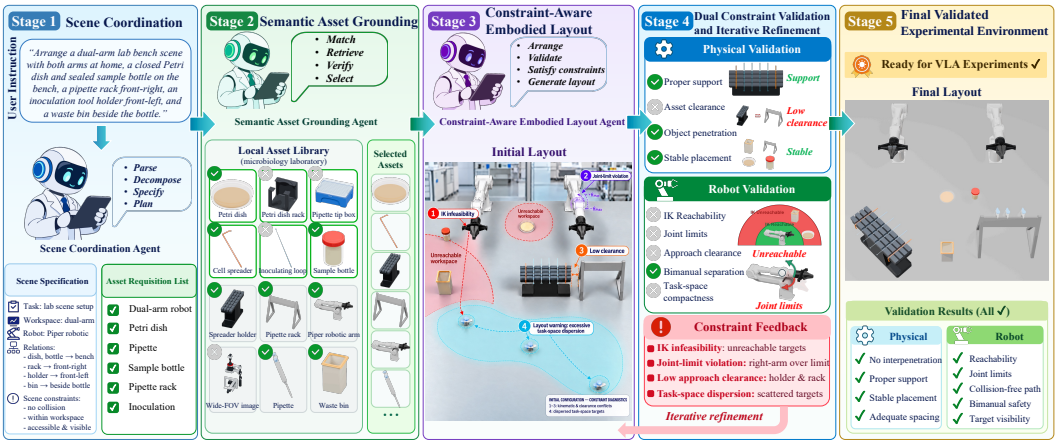


Figure 2: **Automated scene construction.** Agents select assets and propose layouts from instructions. Physical and robot-feasibility checks guide iterative refinement, and validated scenes are used in the experiments.

demonstrations and **Response-augmented** on the same pool plus 100 response demonstrations, using the same pretrained initialization and evaluating at 30k steps. Table 2 supports four comparisons. G1/G2 explores whether a Standard-only policy can switch from normal execution to contamination response when given the disposal instruction, without response demonstrations. The other comparisons assess learned disposal (G2/G4), normal-workflow retention (G1/G3), and generalization to held-out sites (G4/G5).

Normal success requires completion of the tool-specific ordered sequence. Contamination-response success requires disposal after contaminating contact without subsequent sample or agar contact; continued streaking or spreading is a failure. The held-out condition changes the contact location or surface while retaining the tool, workflow step, and required response. Appendix E gives instructions, scoring, and response examples.

4 Constructing Microbiology Environments

Reusable scenes support precise manipulation and distinguish intended from prohibited contact. Construction combines expert asset modeling with automated scene assembly and validation.

4.1 Expert-Built Laboratory Assets

Our library contains **18 assets modeled directly in SolidWorks**, with expert-specified dimensions, component interfaces, and working surfaces. Each asset combines separate visual and collision meshes with a MuJoCo XML (MJCF) specification of mass, inertia, materials, collision settings, and articulated parts. Component geometry defines interfaces such as a plate and its lid; manipulation anchors support motion checks. Working-surface annotations distinguish shaft handling from working-end contact and attach the eligibility state in Equation 2 to the corresponding tool region. Assets are tested individually before assembly; Appendix D gives the inventory and CAD examples.

4.2 Automated Scene Construction

A LangChain-based coordination agent translates experimental instructions into object, relation, and manipulation requirements. An asset-grounding agent selects compatible assets, and a layout agent proposes their poses (Figure 2; prompts in Appendix G). *Physical validation* uses MuJoCo compilation, geometry queries, and simulation to assess support, spacing, penetration, and stability. *Robot-feasibility validation* checks reachability and joint limits through inverse kinematics, scene and inter-arm clearance along planned motions, and target visibility in rendered camera views.

Table 3: **Task-level SR distribution.** Zero and ≥ 80 count tasks with SR $= 0\%$ and SR $\geq 80\%$, respectively.

Level	Model	Mean	Median	Zero	≥ 80
L0 (16)	$\pi_{0.5}$	97.25	99	0	16
	RDT	91.75	100	0	13
	π_0	89.00	97	0	13
L1 (32)	$\pi_{0.5}$	42.56	45	4	5
	RDT	43.75	47	7	5
	π_0	51.81	63	2	8
L2 (12)	$\pi_{0.5}$	7.50	4	4	0
	RDT	24.83	4	5	1
	π_0	1.00	0	8	0

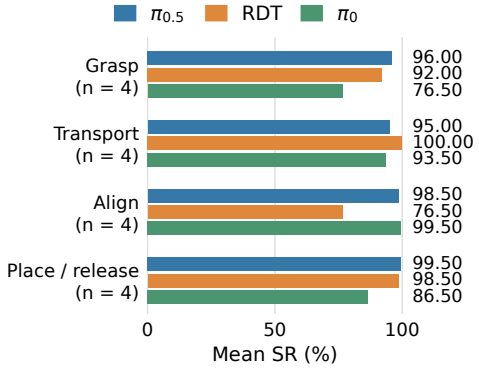


Figure 3: **L0 SR across models.** n counts task variants; details in Appendix B.

Physical and robot critics convert failed checks into object-specific corrections to layouts or asset selections. Scenes passing both checks remain fixed for evaluation, with manifests specifying asset versions, object poses, cameras, task predicates, and validation settings.

5 Experiments

We examine whether VLA policies compose precise microbiology operations and adapt when aseptic constraints require disposal, using nested workflows, cross-level tests, failure analysis, contamination-response studies, and hardware evaluations.

5.1 Experimental setup

We evaluate $\pi_{0.5}$, RDT, and π_0 (Physical Intelligence et al., 2025; Liu et al., 2025; Black et al., 2024). The spreading scene uses a dual-arm setup, with both arms used in L1-21 and L1-23. Workflow training uses 100 demonstrations per L0 or L1 task and 200 per L2 task, matched across models with one multi-task checkpoint per level. Response-study policies train separately on one tool-specific sequence (Section 3.3). Hardware policies train separately per task using 200 teleoperated demonstrations each. All policies train for 30k steps.

We test 60 tasks separately (Lan et al., 2025): 50 trials per model (Tang et al., 2026) (9,000 total). Cross-level evaluation adds 2,550 episodes: each L2 checkpoint runs on L1-01 to L1-14, and each L1 checkpoint runs on L2-02 to L2-04, with 50 trials per cell and no additional training. L1 checkpoints receive the full multi-operation task instruction throughout L2 execution. The contamination-response study uses 50 trials per model–tool–group combination (1,500 total), and hardware evaluation uses 50 per model and task (600 total). Workflow averages weight tasks equally. Appendices B and C provide task-level and cross-level results; Appendix E details the response protocol.

5.2 From atomic actions to multi-operation tasks

Success at the atomic level does not guarantee stable local operation execution. The leading model changes at each level: $\pi_{0.5}$ achieves 97.3% mean SR at L0, π_0 51.8% at L1, and RDT 24.8% at L2 (Table 3). Plate closure illustrates the demands added by composition. Releasing an already-positioned lid succeeds in 100% of episodes for the L0-trained $\pi_{0.5}$ and RDT, yet their L1 checkpoints complete neither plate-closure task (Figure 4). The local operation also requires acquiring and aligning the lid before release. Primitive-level results distinguish model strengths (Figure 3). RDT averages 100% success on transport but 76.5% on alignment; π_0 averages 99.5% on alignment but 76.5% on grasping. $\pi_{0.5}$ achieves 95.0–99.5% across all four families, with all 16 atomic tasks exceeding 80% SR.

Multi-operation task success concentrates in plate handling. Local performance varies sharply within the same workflow (Figure 4): pipette transport succeeds in 92–100% of episodes, whereas

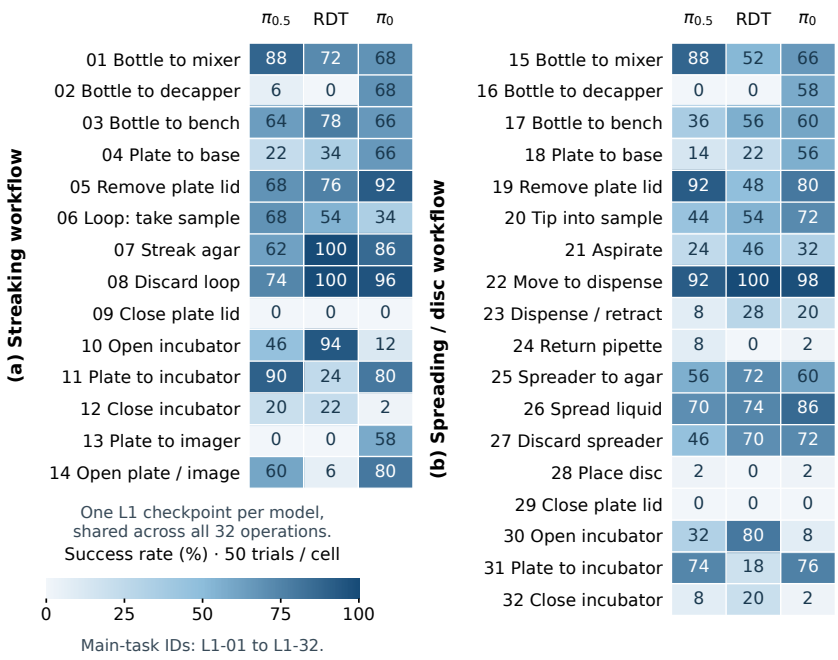


Figure 4: **Success across local operations.** Each cell gives SR (%) over 50 trials using the L1-trained checkpoint. Both plate-closure tasks have zero SR; returning the pipette and placing a disc are also difficult. Appendix B gives task names and intervals.

return to the rack succeeds in only 0–8%. At L2, RDT's mean SR of 24.8% accompanies a median of 4%. Four plate-handling multi-operation tasks account for 138 of its 149 successes (92.6%), averaging 69.0% SR against 2.8% for the other eight multi-operation tasks.

Ordered progress distinguishes incomplete executions. Across the three models, L2-08 (aspiration) has 4.00–6.67% PG and zero SR; L2-10 (spreading) has 44.00–55.33% PG and 0–4% SR (Table 4). On L2-04, RDT reaches 92% PG and 88% SR, while π_0 reaches 43.33% PG with zero SR. PG distinguishes early from later failure; cross-level tests compare isolated and composed execution.

Training level changes success rates. On L1-01 to L1-14, mean-SR rankings change from $\pi_0 > \pi_{0.5} >$ RDT with L1 checkpoints to RDT $> \pi_{0.5} > \pi_0$ with L2 checkpoints (Table 5a). Mean SR falls for all three models, least for RDT and most for π_0. Gains on some tasks are outweighed by losses on others: RDT's L2 checkpoint improves plate closure from zero to 32% SR but reduces loop disposal from 100% to zero (See Appendix C for more details).

Table 4: **Level-2 multi-operation task performance.** SR and PG (%), $N = 50$ episodes per task. Wilson 95% intervals: Table 14. Bold marks the highest nonzero SR per row (ties included).

Multi-operation task	$\pi_{0.5}$		RDT		π_0	
	SR	PG	SR	PG	SR	PG
Workflow A: streak-plate workflow						
L2-01 Mix / open bottle	**4**	10.00	0	3.33	0	2.00
L2-02 Transfer / open plate	22	61.00	**70**	85.00	0	45.00
L2-03 Streak inoculation	0	5.50	**14**	43.00	0	7.00
L2-04 Incubation handoff	12	42.67	**88**	92.00	0	43.33
L2-05 Post-incubation imaging	**26**	63.00	6	53.00	2	25.00
Workflow B: spread-plate / disc workflow						
L2-06 Mix / open bottle	**4**	6.67	0	4.00	0	0.00
L2-07 Transfer / open plate	12	44.00	**52**	68.00	0	36.00
L2-08 Aspirate liquid	0	5.33	0	6.67	0	4.00
L2-09 Dispense liquid	**2**	33.00	0	41.00	**2**	37.00
L2-10 Spread-plate inoculation	0	46.00	2	55.33	**4**	44.00
L2-11 Place disc / close plate	0	8.00	0	9.00	0	8.00
L2-12 Incubation handoff	8	33.33	**66**	80.00	4	51.33

Table 5: **Cross-level checkpoint evaluation.** (a) L1- and L2-trained checkpoints are tested on the same 14 isolated L1 tasks (L1-01–14); entries are mean SR. (b) L1-trained checkpoints follow the full L2-02 instruction to transfer the plate and remove its lid; outcome shares are derived from PG and SR. All entries are percentages, using 50 trials per checkpoint and task without further training.

(a) Same 14 isolated L1 tasks

Model	L1-trained checkpoint	L2-trained checkpoint
$\pi_{0.5}$	47.7	40.4
RDT	47.1	40.9
π_0	57.7	21.9

(b) L1 checkpoints on L2-02

Model	Transfer incomplete	Transfer only	Both steps complete
$\pi_{0.5}$	94	6	0
RDT	32	68	0
π_0	2	98	0

Zero multi-operation task success conceals different execution stages. Under full L2 instructions unseen during L1 training, all L1 checkpoints score zero SR on L2-02–04, jointly testing instruction generalization and operation composition. On L2-02, transfer completion is $r_1 = \text{2PG} - \text{SR}$: 6%, 68%, and 98% for $\pi_{0.5}$, RDT, and π_0 (Table 5b). Most $\pi_{0.5}$ rollouts fail during transfer; most RDT and π_0 rollouts complete transfer but fail at lid removal.

Isolated success gives a different view of multi-operation task execution. Multiplying a checkpoint's isolated SRs gives a reference for multi-operation task completion. The L1-trained π_0 scores 66% on transfer and 92% on lid removal, giving a 60.7% product versus zero multi-operation task SR. Conversely, RDT's L2 checkpoint scores 80%, 42%, and 74% on isolated incubator opening, transfer, and closure: their product is 24.9%, while multi-operation task SR is 88%. At least 88% of multi-operation task rollouts therefore complete transfer, versus 42% in isolation. Isolated performance can overestimate or underestimate workflow completion (full comparisons in Appendix C), connecting operation success to the execution context and states encountered when composing behaviors (Ross et al., 2011; Mandlekar et al., 2020).

5.3 How workflow execution breaks down

We collect 100 failed trajectories per model across L1 and L2 tasks and analyze their causes, identifying three common failure modes (Table 6).

Table 6: **Failure-mode frequencies (%).** 100 failures per model; Other contains the remaining cases.

Failure mode	$\pi_{0.5}$	RDT	π_0
Error accumulation	72	43	58
State ambiguity	6	23	2
Phase confusion	13	28	33
Other	9	6	7

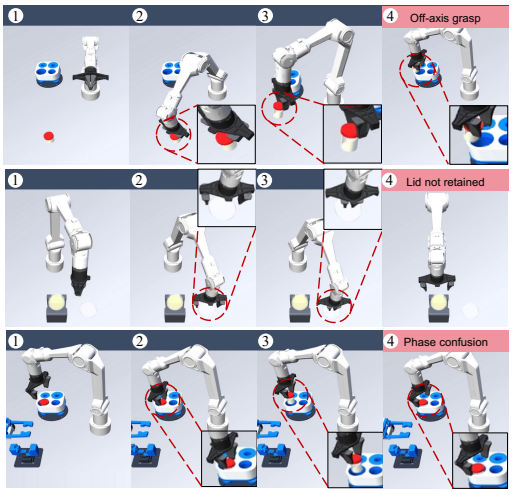


Figure 5: **Workflow failures.** From top to bottom: grasp error propagating to mixer placement, gripper motion without lid retention, and phase confusion near a slot.

Error accumulation. An off-axis bottle grasp causes misalignment during subsequent mixer placement (Figure 5, top). This accounts for 72%, 43%, and 58% of failures for $\pi_{0.5}$, RDT, and π_0, respectively.

State ambiguity. Uncertain grasp or retention leads to repeated attempts or motion without the lid (Figure 5, middle). This accounts for 6%, 23%, and 2% of failures, respectively.

Phase confusion. Similar views across stages trigger the wrong operation, such as retrieval before insertion (Figure 5, bottom). This accounts for 13%, 28%, and 33% of failures, respectively.

Table 7: **Disposal and normal-workflow SR (%).** Group definitions: Table 2. It shows response training improves disposal at seen sites but generalizes poorly. See more details in Appendix C.

Inoculating loop (L1-06/07/08)

Model	G1	G2	G3	G4	G5
$\pi_{0.5}$	22	24	12	100	20
RDT	60	56	58	100	18
π_0	10	18	0	100	0

Spreader (L2-10)

Model	G1	G2	G3	G4	G5
$\pi_{0.5}$	62	0	54	92	0
RDT	94	90	24	90	18
π_0	0	2	0	86	0

5.4 Learning when to discard a tool

Table 7 uses single-sequence policies; Table 4 uses joint L2 training and includes loop plate closure. G1/G2 compares Standard-only normal execution with instructed disposal. Loop-sequence SR rises from 22% to 24% for $\pi_{0.5}$ and from 10% to 18% for π_0, yet $\pi_{0.5}$ falls from 62% to zero on L2-10. The disposal instruction alone does not ensure reliable contamination responses.

Response demonstrations enable disposal at seen sites. With Standard-only training, disposal SR ranges from 18% to 56% for loops and 0% to 90% for spreaders. Adding response demonstrations brings all loop models to 100% and spreader models to 86–92%. RDT already achieves 90% on spreaders before response training; the largest gains come from $\pi_{0.5}$ and π_0, whose spreader disposal improves by 92 and 84 percentage points, respectively.

Normal execution and disposal follow different patterns. For loops, response training lowers normal-workflow SR for every model, even as seen-site disposal reaches 100%. For spreaders, $\pi_{0.5}$ falls from 62% to 54% normal SR, while RDT falls from 94% to 24%. π_0 completes neither normal workflow after response training despite high disposal success. Learning the required response thus interacts differently with ordinary tool use across models and workflows. For spreaders, $\pi_{0.5}$ rises from 76% to 84.67% PG, while RDT falls from 96% to 44% (Appendix C).

Unseen contact sites expose a generalization gap. Held-out loop disposal reaches 20% for $\pi_{0.5}$, 18% for RDT, and 0% for π_0; spreader disposal reaches 18% for RDT and zero for the others. Changing the contact site sharply reduces success despite the same required action, showing strong sensitivity to where prohibited contact occurs.

5.5 Laboratory operations on hardware

Table 8: **Physical task SR (%).** 50 trials per model and task; bold marks the highest SR.

Task	$\pi_{0.5}$	RDT	π_0
Pipette immersion	**64**	24	30
Lid removal	**54**	4	4
Spreader grasping	**60**	24	28
Streaking	**52**	28	20

We choose four tasks for hardware evaluation because their L1 counterparts have high simulation SR and they represent microbiology tool acquisition, plate handling, and tool–agar interaction (Appendix F). $\pi_{0.5}$ leads with 52–64% SR, while RDT and π_0 achieve 4–30%, with lid removal particularly difficult (Table 8).

The ranking changes across settings: RDT leads simulated streaking at 100%, followed by π_0 at 86% and $\pi_{0.5}$ at 62%; hardware policies trained on physical demonstrations achieve 28%, 20%, and 52%, respectively. Physical interaction adds a distinct test of local execution.

6 Conclusion

We introduced MicroBio-Bench, a reusable simulation environment and benchmark for microbiology manipulation with aseptic constraints. It enables repeated training and evaluation without consuming laboratory supplies and makes contact-dependent tool eligibility explicit. Evaluations of $\pi_{0.5}$, RDT, and π_0 across 60 nested tasks and a five-group contamination-response study reveal difficulties in executing operations precisely and reliably, composing them into workflows, and generalizing disposal to new contact sites. Cross-level tests highlight the gap between isolated operations and workflow execution, while failure analysis identifies error accumulation, state ambiguity, and phase confusion. Together, the benchmark and these findings provide a testbed and directions for improving VLA performance in microbiology laboratories.

AI USE STATEMENT

In this work, generative AI tools were used to assist with manuscript preparation, presentation, and literature retrieval and discovery. For paper preparation, we used large language models to assist with drafting and translation, improve the clarity, concision, and grammar of the text, suggest alternative phrasings, and help format tables and figures in LaTeX.

For literature retrieval and discovery, we used AI tools to help identify potentially relevant prior work on vision-language-action models, robotic laboratory automation, and benchmarks for manipulation. AI tools also assisted with methodological critique, preparation of analysis code, and visualization of experimental results.

REPRODUCIBILITY STATEMENT

We provide the constructed digital assets and demonstration videos of L1 tasks, L2 tasks, contamination recovery, and real-world evaluations in the supplementary material. The Appendix documents the task settings and evaluation protocols (Appendices A, E, and F) and the training and inference details (Appendix H).

REFERENCES

Alibaba Cloud. Qwen3.7-Flash. Alibaba Cloud Model Studio documentation, 2026. URL https://www.alibabacloud.com/help/en/model-studio/qwen3-7-flash.

Kevin Black, Noah Brown, Danny Driess, Adnan Esmail, Michael Equi, Chelsea Finn, Niccolo Fusai, Lachy Groom, Karol Hausman, Brian Ichter, Szymon Jakubczak, Tim Jones, Liyiming Ke, Sergey Levine, Adrian Li-Bell, Mohith Mothukuri, Suraj Nair, Karl Pertsch, Lucy Xiaoyang Shi, James Tanner, Quan Vuong, Anna Walling, Haohuan Wang, and Ury Zhilinsky. π_0: A vision-language-action flow model for general robot control. *arXiv preprint arXiv:2410.24164*, 2024. URL https://arxiv.org/abs/2410.24164.

Daniil A. Boiko, Robert MacKnight, Ben Kline, and Gabe Gomes. Autonomous chemical research with large language models. *Nature*, 624:570–578, 2023. doi: 10.1038/s41586-023-06792-0. URL https://www.nature.com/articles/s41586-023-06792-0.

Anthony Brohan, Noah Brown, Justice Carbajal, Yevgen Chebotar, Xi Chen, Krzysztof Choromanski, Tianli Ding, Danny Driess, Avinava Dubey, Chelsea Finn, Pete Florence, Chuyuan Fu, Montse Gonzalez Arenas, Keerthana Gopalakrishnan, Kehang Han, Karol Hausman, Alexander Herzog, Jasmine Hsu, Brian Ichter, Alex Irpan, Nikhil Joshi, Ryan Julian, Dmitry Kalashnikov, Yuheng Kuang, Isabel Leal, Lisa Lee, Tsang-Wei Edward Lee, Sergey Levine, Yao Lu, Henryk Michalewski, Igor Mordatch, Karl Pertsch, Kanishka Rao, Krista Reymann, Michael Ryoo, Grecia Salazar, Pannag Sanketi, Pierre Sermanet, Jaspiar Singh, Anikait Singh, Radu Soricut, Huong Tran, Vincent Vanhoucke, Quan Vuong, Ayzaan Wahid, Stefan Welker, Paul Wohlhart, Jialin Wu, Fei Xia, Ted Xiao, Peng Xu, Sichun Xu, Tianhe Yu, and Brianna Zitkovich. RT-2: Vision-language-action models transfer web knowledge to robotic control. *arXiv preprint arXiv:2307.15818*, 2023. URL https://arxiv.org/abs/2307.15818.

Benjamin Burger, Phillip M. Maffettone, Vladimir V. Gusev, Catherine M. Aitchison, Yang Bai, Xiaoyan Wang, Xiaobo Li, Ben M. Alston, Buyi Li, Rob Clowes, Nicola Rankin, Brandon Harris, Reiner Sebastian Sprick, and Andrew I. Cooper. A mobile robotic chemist. *Nature*, 583:237–241, 2020. doi: 10.1038/s41586-020-2442-2. URL https://www.nature.com/articles/s41586-020-2442-2.

Tianxing Chen, Yue Chen, Zixuan Li, Junyuan Tang, Kailun Su, Weijie Wan, Baijun Chen, Haoran Lu, Haowen Yan, Honghao Su, et al. RoboDojo: A unified sim-and-real benchmark for comprehensive evaluation of generalist robot manipulation policies. *arXiv preprint arXiv:2607.04434*, 2026. URL https://arxiv.org/abs/2607.04434.

Chengyue Huang, Khang Vo Huynh, Sebastian Elbaum, Zsolt Kira, and Lu Feng. SafeManip: A property-driven benchmark for temporal safety evaluation in robotic manipulation. *arXiv preprint arXiv:2605.12386*, 2026. URL https://arxiv.org/abs/2605.12386.

Moo Jin Kim, Karl Pertsch, Siddharth Karamcheti, Ted Xiao, Ashwin Balakrishna, Suraj Nair, Rafael Rafailov, Ethan P Foster, Pannag R Sanketi, Quan Vuong, Thomas Kollar, Benjamin Burchfiel, Russ Tedrake, Dorsa Sadigh, Sergey Levine, Percy Liang, and Chelsea Finn. OpenVLA: An open-source vision-language-action model. In *Proceedings of The 8th Conference on Robot Learning*, volume 270 of *Proceedings of Machine Learning Research*, pp. 2679–2713. PMLR, 2025. URL https://proceedings.mlr.press/v270/kim25c.html.

Zhiqian Lan, Yuxuan Jiang, Ruiqi Wang, Xuanbing Xie, Rongkui Zhang, Yicheng Zhu, Peihang Li, Tianshuo Yang, Tianxing Chen, Haoyu Gao, Xiaokang Yang, Xuelong Li, Hongyuan Zhang, Yao Mu, and Ping Luo. AutoBio: A simulation and benchmark for robotic automation in digital biology laboratory. *arXiv preprint arXiv:2505.14030*, 2025. URL https://arxiv.org/abs/2505.14030.

Rui Li, Zixuan Hu, Wenxi Qu, Jinouwen Zhang, Zhenfei Yin, Sha Zhang, Xuantuo Huang, Hanqing Wang, Tai Wang, Jiangmiao Pang, Wanli Ouyang, Lei Bai, Wangmeng Zuo, Ling-Yu Duan, Dongzhan Zhou, and Shixiang Tang. LabUtopia: High-fidelity simulation and hierarchical benchmark for scientific embodied agents. *arXiv preprint arXiv:2505.22634*, 2025. URL https://arxiv.org/abs/2505.22634. Accepted to NeurIPS 2025 Datasets and Benchmarks Track.

Shoujie Li, Yan Huang, Changqing Guo, Tong Wu, Jiawei Zhang, Linrui Zhang, and Wenbo Ding. Chemistry3D: Robotic interaction benchmark for chemistry experiments. *arXiv preprint arXiv:2406.08160*, 2024. URL https://arxiv.org/abs/2406.08160.

Bo Liu, Yifeng Zhu, Chongkai Gao, Yihao Feng, Qiang Liu, Yuke Zhu, and Peter Stone. LIBERO: Benchmarking knowledge transfer for lifelong robot learning. In *Advances in Neural Information Processing Systems*, 2023a. URL https://arxiv.org/abs/2306.03310.

Songming Liu, Lingxuan Wu, Bangguo Li, Hengkai Tan, Huayu Chen, Zhengyi Wang, Ke Xu, Hang Su, and Jun Zhu. RDT-1B: A diffusion foundation model for bimanual manipulation. In *International Conference on Learning Representations*, 2025. URL https://arxiv.org/abs/2410.07864.

Zeyi Liu, Arpit Bahety, and Shuran Song. REFLECT: Summarizing robot experiences for failure explanation and correction. In *Proceedings of The 7th Conference on Robot Learning*, volume 229 of *Proceedings of Machine Learning Research*, pp. 3468–3484. PMLR, 2023b. URL https://proceedings.mlr.press/v229/liu23g.html.

Zhe Liu, Huanbo Jin, Zhaohui Du, Zhe Wang, Dongzhan Zhou, Minting Pan, He Xu, Peijia Li, Jiaming Gu, Quan Lu, Qi Wang, Bin Ji, and Ting Xiao. Pipette: An embodied simulation platform, benchmark, and data-efficient augmentation framework for wet-lab robotics. *arXiv preprint arXiv:2606.12936*, 2026. URL https://arxiv.org/abs/2606.12936.

Ajay Mandlekar, Danfei Xu, Roberto Martín-Martín, Silvio Savarese, and Li Fei-Fei. GTI: Learning to generalize across long-horizon tasks from human demonstrations. In *Proceedings of Robotics: Science and Systems*, 2020. doi: 10.15607/RSS.2020.XVI.061. URL https://www.roboticsproceedings.org/rss16/p061.html.

Ajay Mandlekar, Danfei Xu, Josiah Wong, Soroush Nasiriany, Chen Wang, Rohun Kulkarni, Li Fei-Fei, Silvio Savarese, Yuke Zhu, and Roberto Martín-Martín. What matters in learning from offline human demonstrations for robot manipulation. In *Proceedings of the 5th Conference on Robot Learning*, volume 164 of *Proceedings of Machine Learning Research*, pp. 1678–1690. PMLR, 2022. URL https://proceedings.mlr.press/v164/mandlekar22a.html.

Oier Mees, Lukas Hermann, Erick Rosete-Beas, and Wolfram Burgard. CALVIN: A benchmark for language-conditioned policy learning for long-horizon robot manipulation tasks. *IEEE Robotics and Automation Letters*, 2022. URL https://arxiv.org/abs/2112.03227.

Octo Model Team, Dibya Ghosh, Homer Walke, Karl Pertsch, Kevin Black, Oier Mees, Sudeep Dasari, Joey Hejna, Charles Xu, Jianlan Luo, Tobias Kreiman, You Liang Tan, Lawrence Yunliang Chen, Pannag Sanketi, Quan Vuong, Ted Xiao, Dorsa Sadigh, Chelsea Finn, and Sergey Levine. Octo: An open-source generalist robot policy. In *Proceedings of Robotics: Science and Systems*, 2024. URL https://arxiv.org/abs/2405.12213.

Physical Intelligence, Kevin Black, Noah Brown, James Darpinian, Karan Dhabalia, Danny Driess, Adnan Esmail, Michael Equi, Chelsea Finn, Niccolo Fusai, Manuel Y. Galliker, Dibya Ghosh, Lachy Groom, Karol Hausman, Brian Ichter, Szymon Jakubczak, Tim Jones, Liyiming Ke, Devin LeBlanc, Sergey Levine, Adrian Li-Bell, Mohith Mothukuri, Suraj Nair, Karl Pertsch, Allen Z. Ren, Lucy Xiaoyang Shi, Laura Smith, Jost Tobias Springenberg, Kyle Stachowicz, James Tanner, Quan Vuong, Homer Walke, Anna Walling, Haohuan Wang, Lili Yu, and Ury Zhilinsky. $\pi_{0.5}$: A vision-language-action model with open-world generalization. *arXiv preprint arXiv:2504.16054*, 2025. URL https://arxiv.org/abs/2504.16054.

Stephane Ross, Geoffrey Gordon, and Drew Bagnell. A reduction of imitation learning and structured prediction to no-regret online learning. In *Proceedings of the Fourteenth International Conference on Artificial Intelligence and Statistics*, volume 15 of *Proceedings of Machine Learning Research*, pp. 627–635. PMLR, 2011. URL https://proceedings.mlr.press/v15/ross11a.html.

Erin R. Sanders. Aseptic laboratory techniques: Plating methods. *Journal of Visualized Experiments*, (63):e3064, 2012. doi: 10.3791/3064. URL https://www.jove.com/t/3064/aseptic-laboratory-techniques-plating-methods.

Zhipeng Tang, Sihang Chen, Sha Zhang, Peihao Yang, Yan Liu, Wentao Zhao, Xinrui Liu, Rui Huang, Wensheng Du, Yuting Huang, Jiajun Deng, Lidian Wang, Yuan Zhang, and Yanyong Zhang. LabDex: A hierarchical benchmark for dexterous manipulation in laboratories. *arXiv preprint arXiv:2608.18618*, 2026. URL https://arxiv.org/abs/2608.18618.

Yufei Wang, Zhou Xian, Feng Chen, Tsun-Hsuan Wang, Yian Wang, Katerina Fragkiadaki, Zackory Erickson, David Held, and Chuang Gan. RoboGen: Towards unleashing infinite data for automated robot learning via generative simulation. *arXiv preprint arXiv:2311.01455*, 2023. URL https://arxiv.org/abs/2311.01455.

Yuhan Wu, Zhao Jin, Tao Li, Yuheng Zhang, Zhengping Che, Jian Tang, Zhichao Wang, Shuo Wang, Jun Jiang, Xiaobo Li, Yanyong Zhang, and Yan Xia. Labimus: A simulation and benchmark for humanoid dexterous manipulation in chemical laboratory. *arXiv preprint arXiv:2606.31037*, 2026. URL https://arxiv.org/abs/2606.31037.

A TASK CATALOG AND NESTING MAP

L0 suffixes distinguish task instances within the four primitive families. The 32 main L1 operations are numbered consecutively L1-01 to L1-32. Operations appearing in both workflow families are assigned separate IDs for the different workflow contexts.

Table 9: L2 multi-operation tasks and their ordered L1 operations. Each local operation uses the applicable transitions involving grasping, transport, alignment, and release.

Multi-operation task	Description	Ordered local operations
L2-01	Mix and open sample bottle	L1-01, 02, 03
L2-02	Transfer and open plate	L1-04, 05
L2-03	Streak inoculation	L1-06, 07, 08, 09
L2-04	Incubation handoff	L1-10, 11, 12
L2-05	Post-incubation imaging	L1-13, 14
L2-06	Mix and open sample bottle	L1-15, 16, 17
L2-07	Transfer and open plate	L1-18, 19
L2-08	Aspirate liquid	L1-20, 21, 22
L2-09	Dispense liquid and return pipette	L1-23, 24
L2-10	Spread-plate inoculation and tool disposal	L1-25, 26, 27
L2-11	Place disc and close plate	L1-28, 29
L2-12	Incubation handoff	L1-30, 31, 32

The loop workflow comprises L2-01–05. The spreading workflow comprises L2-06–12, with L2-11 as the optional disc-placement branch. The primary task set covers manipulation through incubation handoff and the two L1 imaging operations. Device-related tasks distinguish object placement from instrument-state transitions.

A.1 L1 SUCCESS CRITERIA

Each L1 operation is scored by the task-specific reference geometry and predicate in Table 11. Table 10 defines the seven predicate families. A successful episode satisfies every condition of its assigned predicate; an episode is unsuccessful if any required condition remains unmet at termination. Position tolerances and path lengths are in millimeters. For liquid insertion, signed height is measured relative to the liquid surface, with negative values below the surface.

Table 10: Predicate families used to determine L1 completion. Numeric parameters are supplied for each operation in Table 11.

Predicate	Required conditions
P: placement	The object reference point reaches the target tolerance, the target surface supports the object, and the gripper releases it. Surface-based placements use the specified support interface.
O: removal	The removed object's reference point reaches the staging tolerance, the staging surface supports it, and the gripper releases it.
C: closure	The lid's inner surface contacts the base and the gripper releases the lid. Incubator closure also requires the stated center-position tolerance.
H: held-tool positioning	Both finger pads hold the tool and its working point enters the specified target region.
A: actuation	The button travels at least the specified distance.
T: trajectory	The working end contacts the target surface, completes at least the specified path length, and exits as required by the operation.
D: disposal	The object enters the waste container, obtains support inside it, and is released by the gripper.

Table 11: Success criteria for all L1 operations. Each geometry entry lists the object reference followed by its target. P and O parameters are position tolerances; C(15) adds a 15 mm center tolerance; A parameters are minimum button travel; T parameters are minimum contact-path length. H intervals specify the working-point target region relative to the named surface.

Task	Operation	Object reference; target	Predicate (mm)
L1-01	Bottle to mixer	Bottle-bottom center; mixer-slot bottom center	P(5)
L1-02	Bottle to decapper	Bottle-bottom center; decapper-base center	P(5)
L1-03	Bottle to workbench	Bottle-bottom center; workbench placement point	P(10)
L1-04	Plate to operation base	Plate-bottom center; operation-base center	P(10)
L1-05	Remove plate lid	Lid center; lid staging point 1	O(20)
L1-06	Loop: take sample	Loop end; liquid surface	H: $[-35, 0]$
L1-07	Streak agar	Loop end; agar surface	T(25)
L1-08	Discard loop	Loop end; waste-container bottom	D
L1-09	Close plate lid	Lid inner surface; plate-base upper surface	C
L1-10	Open incubator	Incubator-lid center; lid staging point	O(15)
L1-11	Plate to incubator	Plate-bottom center; incubator floor	P: support
L1-12	Close incubator	Lid center; incubator-body center	C(15)
L1-13	Plate to imager	Plate-bottom center; imaging-tray center	P(15)
L1-14	Open plate for imaging	Plate-lid center; lid staging point 2	O(20)
L1-15	Bottle to mixer	Bottle-bottom center; mixer-slot bottom center	P(5)
L1-16	Bottle to decapper	Bottle-bottom center; decapper-base center	P(5)
L1-17	Bottle to workbench	Bottle-bottom center; workbench placement point	P(10)
L1-18	Plate to operation base	Plate-bottom center; operation-base center	P(10)
L1-19	Remove plate lid	Lid center; lid staging point 1	O(20)
L1-20	Pipette tip into sample	Pipette-tip end; liquid surface	H: $[-35, 0]$
L1-21	Aspirate with second arm	Second-arm end effector; pipette button	A(5)
L1-22	Move to dispensing position	Pipette-tip end; agar surface	H: $[2, 30]$
L1-23	Dispense and retract	Second-arm end effector; pipette button	A(5)
L1-24	Return pipette to rack	Pipette rear surface; rack support surface	P: support
L1-25	Spreader to agar	Spreader working end; agar surface	H: $[5, 35]$
L1-26	Spread liquid	Spreader working end; agar surface	T(30)
L1-27	Discard spreader	Spreader working end; waste-container bottom	D
L1-28	Place antibiotic disc	Disc-bottom center; agar center	P(30)
L1-29	Close plate lid	Lid inner surface; plate-base upper surface	C
L1-30	Open incubator	Incubator-lid center; lid staging point	O(15)
L1-31	Plate to incubator	Plate-bottom center; incubator floor	P: support
L1-32	Close incubator	Lid center; incubator-body center	C(15)

A.2 L1 TASK SEQUENCES

Figures 6–13 show 23 local-operation sequences spanning bottle preparation, plate handling, inoculation, pipetting, incubation, and imaging. Each row follows the four-frame layout of Figure 5: initialization, two intermediate stages, and termination. Red dashed circles and linked insets identify the manipulation interfaces. Shared bottle, plate, and incubator operations also occur in the dual-arm spreading workflow, with the corresponding task IDs listed in Appendix A.

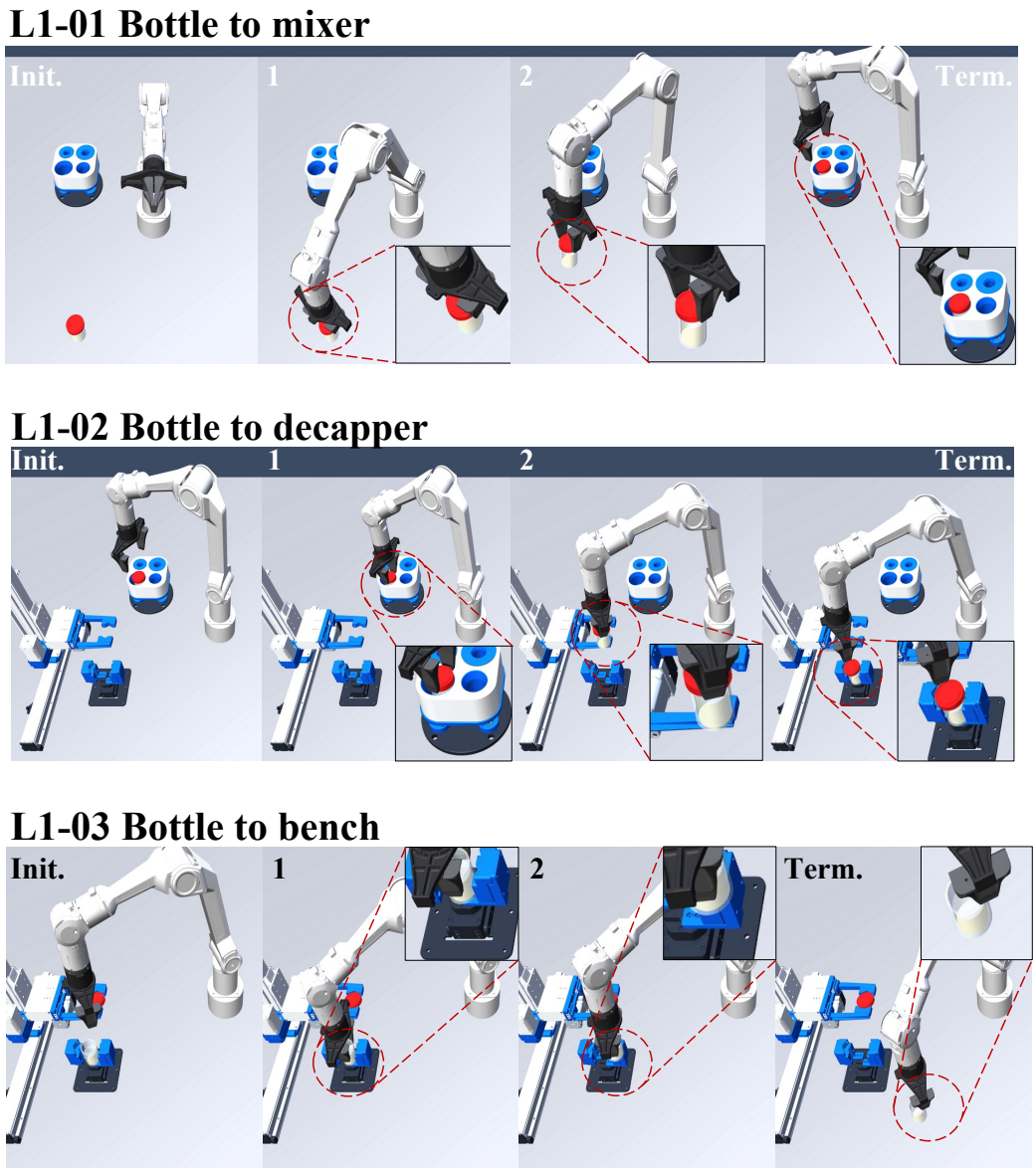


Figure 6: **Bottle preparation.** The bottle moves from the workbench to the mixer, from the mixer to the decapper, and from the decapper back to the workbench. Insets show the grasp and bottle–fixture interfaces.

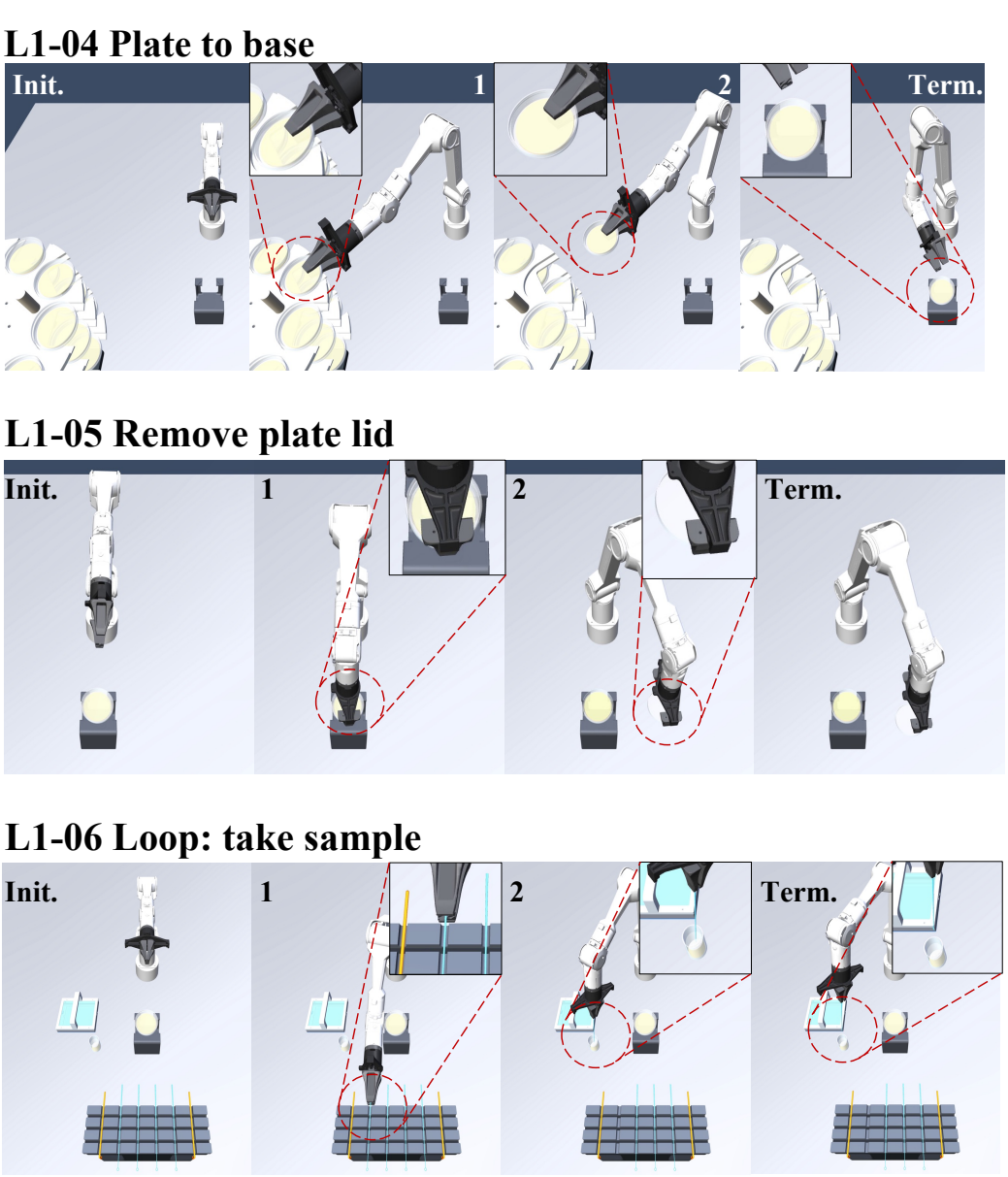


Figure 7: **Plate preparation and loop sampling.** The plate moves from its rack to the operation base, its lid moves to the staging area, and the loop is acquired and inserted into the sample. Insets show plate retention, lid separation, and the loop working end.

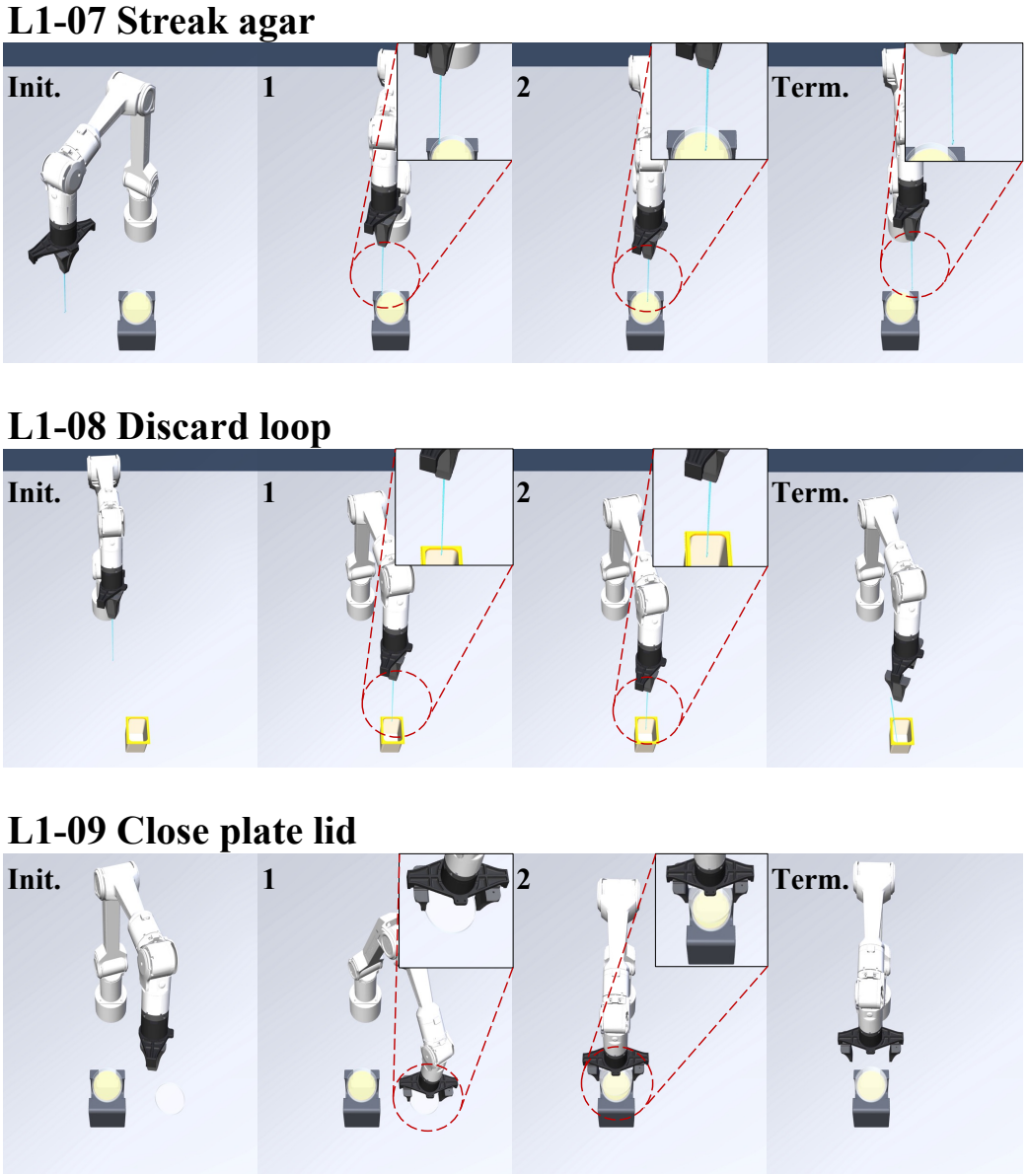


Figure 8: **Streaking, disposal, and plate closure.** The loop traverses the agar, moves into the waste container for release, and the plate lid returns to the plate. The highlighted regions track tool–agar contact, disposal, and lid alignment.

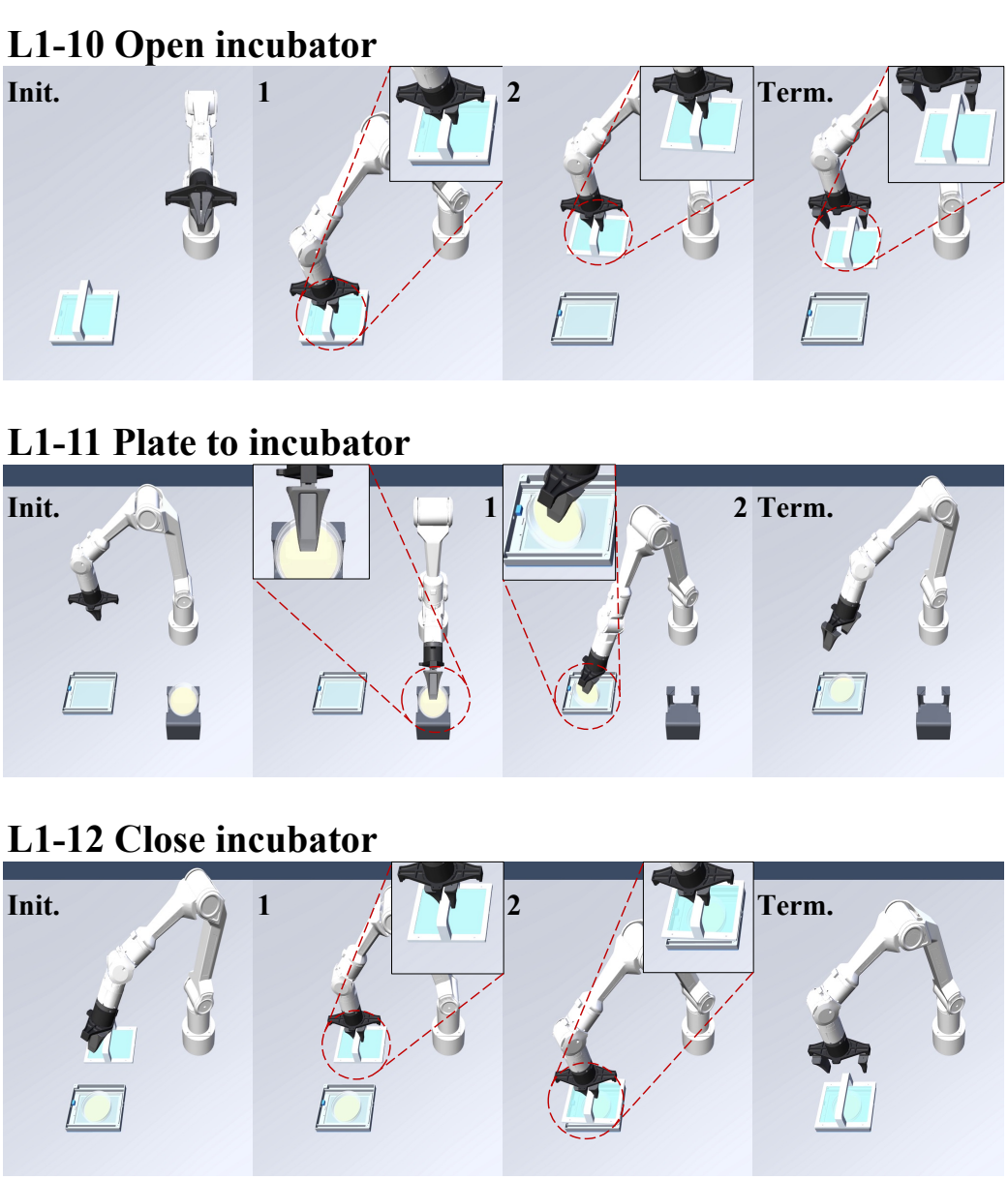


Figure 9: **Incubation handoff.** The incubator lid moves to its staging area, the plate enters the incubator, and the lid returns to the incubator body. Insets identify the lid grasp and plate-support interfaces.

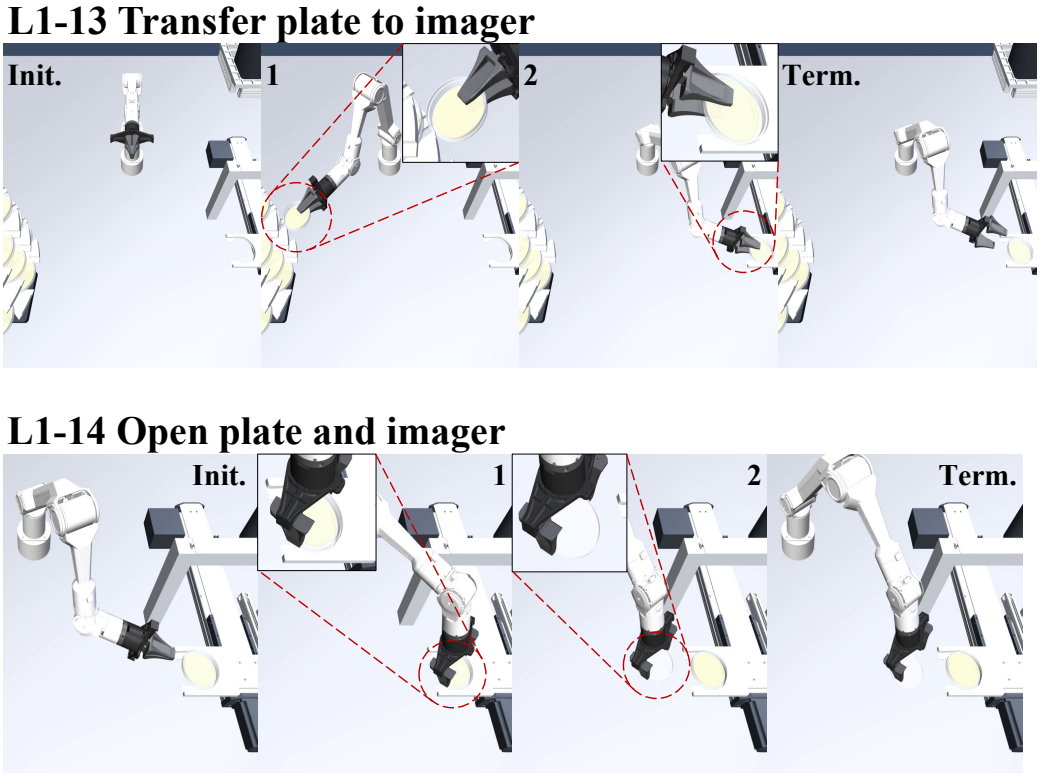


Figure 10: **Post-incubation imaging.** The plate moves from the rack to the imaging tray, followed by lid removal to expose the plate for imaging. Insets show acquisition of the plate and separation of its lid.

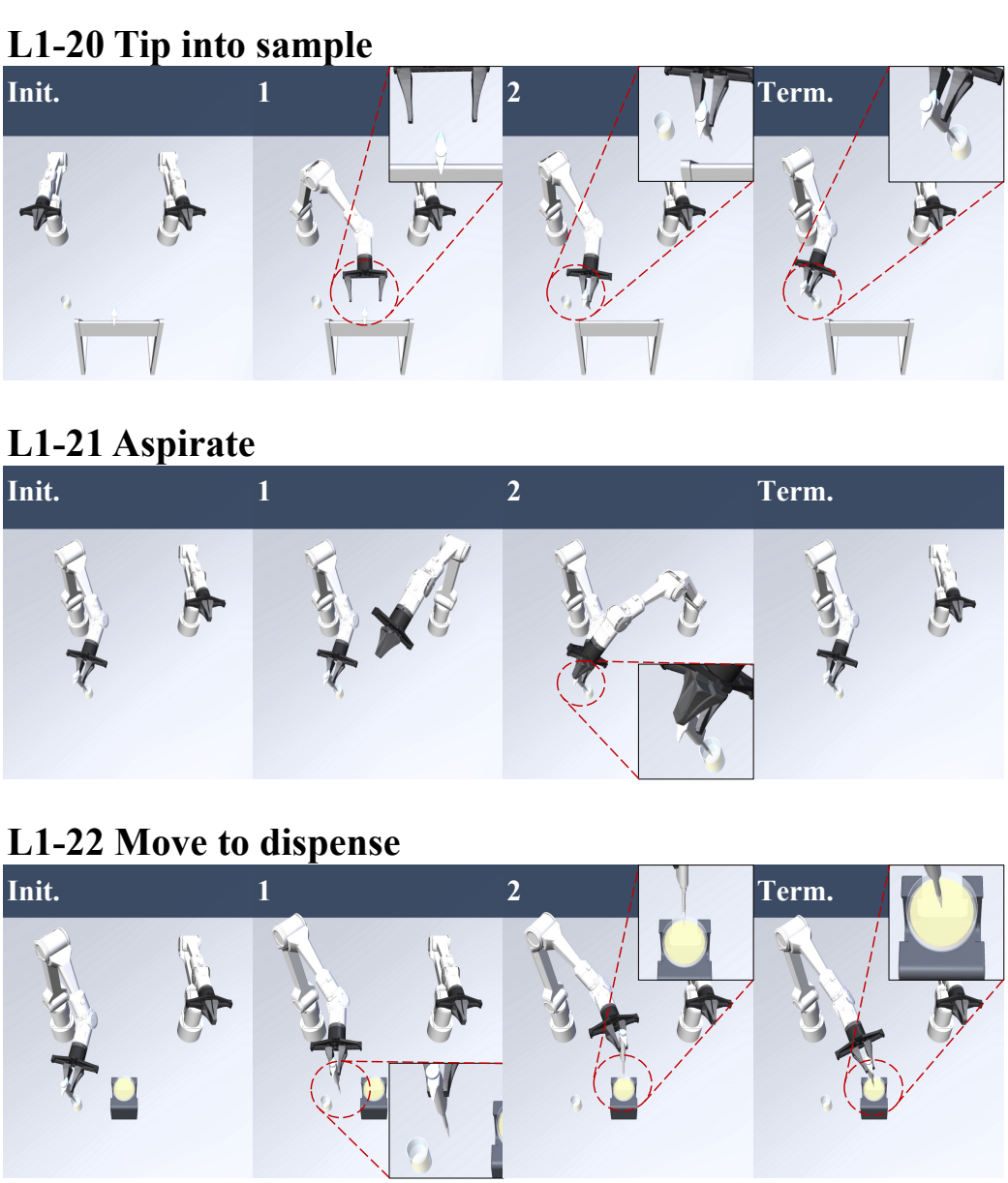


Figure 11: **Pipette acquisition, aspiration, and transport.** One arm acquires the pipette and inserts its tip into the sample, the second arm actuates the button for aspiration, and the pipette moves to the dispensing position above the plate.

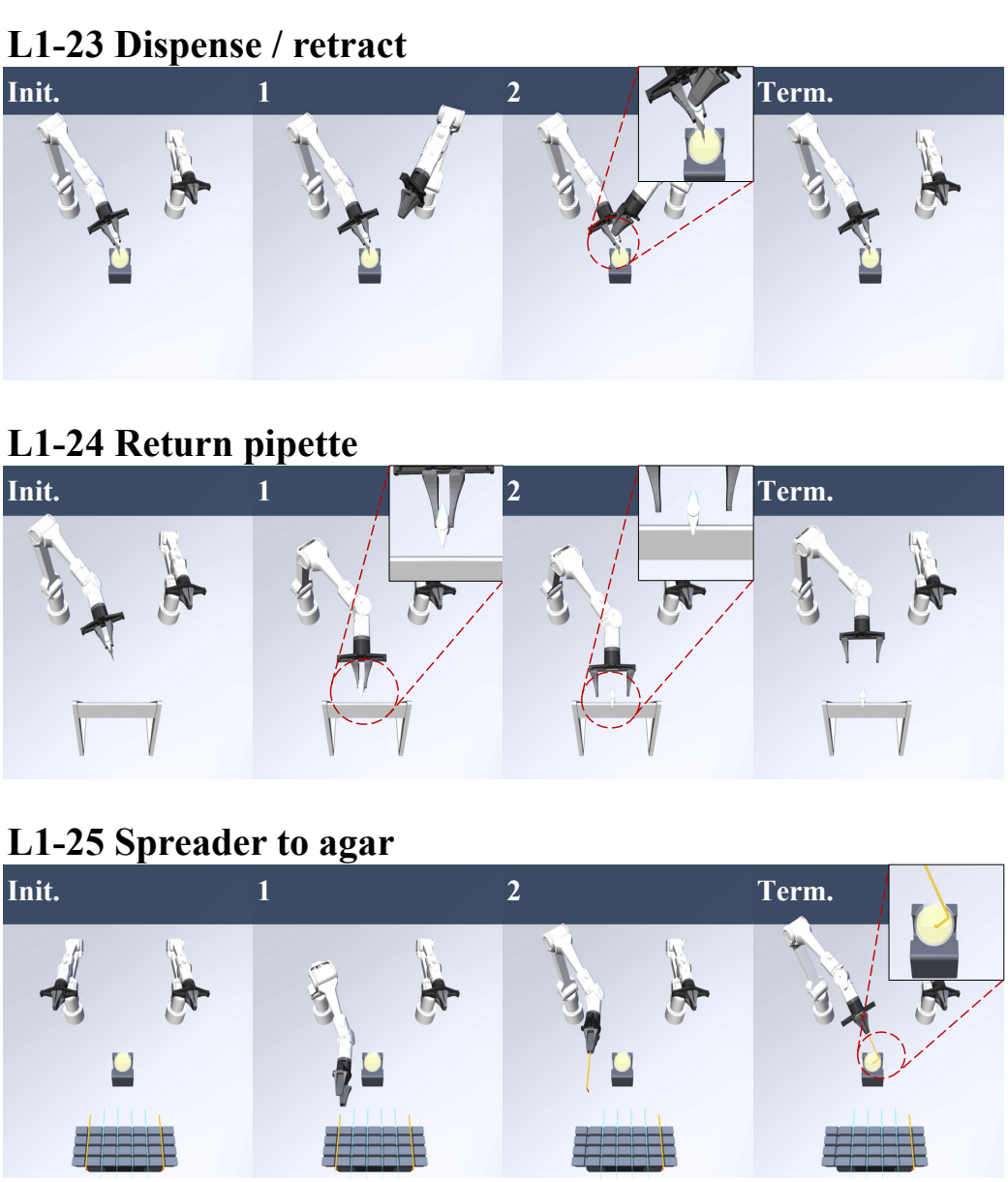


Figure 12: **Dispensing, pipette return, and spreader acquisition.** The second arm actuates the pipette for dispensing and retracts, the pipette returns to its rack, and the spreader moves from the tool holder to the agar region.

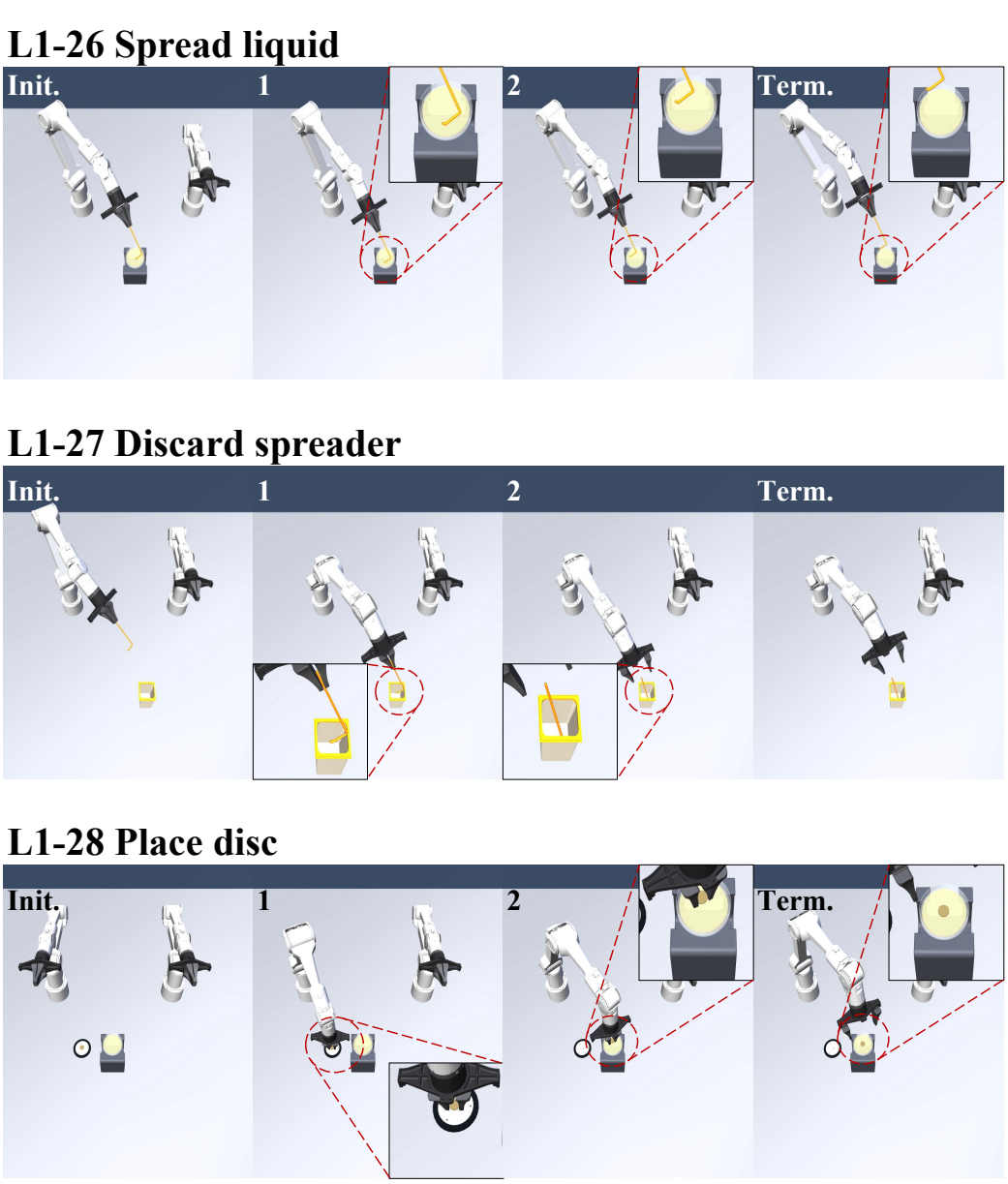


Figure 13: **Spreading, disposal, and disc placement.** The spreader traverses the agar and then moves into the waste container. The disc-placement sequence shows acquisition, transfer, and placement on the agar surface.

B TASK-LEVEL RESULTS

Tables 12–14 report every task instance. Because each task is evaluated in 50 trials, the numerical value of its success rate in percent equals twice the number of successful episodes. Per-task Wilson intervals are computed as

$$\frac{\hat{p} + z^2/(2n) \;\pm\; z\sqrt{\hat{p}(1-\hat{p})/n + z^2/(4n^2)}}{1 + z^2/n}, \qquad n = 50, \quad z = 1.959964. \tag{3}$$

Intervals use binomial episode outcomes for each evaluated checkpoint. PG is the mean completed ordered-subtask fraction (%).

Table 12: All L0 task success rates (%) with per-task Wilson 95% intervals.

Task	Operation	$\pi_{0.5}$	RDT	π_0
L0-01a	Grasp spreader	88 [76.2, 94.4]	76 [62.6, 85.7]	78 [64.8, 87.2]
L0-01b	Grasp inoculating loop	100 [92.9, 100.0]	98 [89.5, 99.6]	88 [76.2, 94.4]
L0-01c	Grasp plate lid	96 [86.5, 98.9]	94 [83.8, 97.9]	92 [81.2, 96.8]
L0-01d	Grasp incubator lid	100 [92.9, 100.0]	100 [92.9, 100.0]	48 [34.8, 61.5]
L0-02a	Move sample bottle to mixer	92 [81.2, 96.8]	100 [92.9, 100.0]	86 [73.8, 93.0]
L0-02b	Move plate to imaging slot	96 [86.5, 98.9]	100 [92.9, 100.0]	92 [81.2, 96.8]
L0-02c	Move loop to plate	100 [92.9, 100.0]	100 [92.9, 100.0]	100 [92.9, 100.0]
L0-02d	Move incubator lid to base	92 [81.2, 96.8]	100 [92.9, 100.0]	96 [86.5, 98.9]
L0-03a	Align bottle with decapper	96 [86.5, 98.9]	62 [48.2, 74.1]	100 [92.9, 100.0]
L0-03b	Align spreader with agar	100 [92.9, 100.0]	64 [50.1, 75.9]	100 [92.9, 100.0]
L0-03c	Align plate with operation base	100 [92.9, 100.0]	100 [92.9, 100.0]	100 [92.9, 100.0]
L0-03d	Align bottle with mixer slot	98 [89.5, 99.6]	80 [67.0, 88.8]	98 [89.5, 99.6]
L0-04a	Place bottle on workbench	98 [89.5, 99.6]	94 [83.8, 97.9]	100 [92.9, 100.0]
L0-04b	Discard loop	100 [92.9, 100.0]	100 [92.9, 100.0]	100 [92.9, 100.0]
L0-04c	Place plate in incubator	100 [92.9, 100.0]	100 [92.9, 100.0]	100 [92.9, 100.0]
L0-04d	Replace plate lid	100 [92.9, 100.0]	100 [92.9, 100.0]	46 [33.0, 59.6]

Table 13: All 32 L1 task success rates (%) with per-task Wilson 95% intervals.

Task	Operation	$\pi_{0.5}$	RDT	π_0
L1-01	Transfer bottle to mixer	88 [76.2, 94.4]	72 [58.3, 82.5]	68 [54.2, 79.2]
L1-02	Transfer bottle to decapper	6 [2.1, 16.2]	0 [0.0, 7.1]	68 [54.2, 79.2]
L1-03	Transfer open bottle to workbench	64 [50.1, 75.9]	78 [64.8, 87.2]	66 [52.2, 77.6]
L1-04	Transfer plate from rack to base	22 [12.8, 35.2]	34 [22.4, 47.8]	66 [52.2, 77.6]
L1-05	Remove plate lid	68 [54.2, 79.2]	76 [62.6, 85.7]	92 [81.2, 96.8]
L1-06	Grasp loop and sample liquid	68 [54.2, 79.2]	54 [40.4, 67.0]	34 [22.4, 47.8]
L1-07	Streak agar surface	62 [48.2, 74.1]	100 [92.9, 100.0]	86 [73.8, 93.0]
L1-08	Discard loop	74 [60.4, 84.1]	100 [92.9, 100.0]	96 [86.5, 98.9]
L1-09	Close plate lid	0 [0.0, 7.1]	0 [0.0, 7.1]	0 [0.0, 7.1]
L1-10	Open incubator lid	46 [33.0, 59.6]	94 [83.8, 97.9]	12 [5.6, 23.8]
L1-11	Transfer plate into incubator	90 [78.6, 95.7]	24 [14.3, 37.4]	80 [67.0, 88.8]
L1-12	Close incubator lid	20 [11.2, 33.0]	22 [12.8, 35.2]	2 [0.4, 10.5]
L1-13	Transfer plate to imager	0 [0.0, 7.1]	0 [0.0, 7.1]	58 [44.2, 70.6]
L1-14	Open plate and image	60 [46.2, 72.4]	6 [2.1, 16.2]	80 [67.0, 88.8]
L1-15	Transfer bottle to mixer	88 [76.2, 94.4]	52 [38.5, 65.2]	66 [52.2, 77.6]
L1-16	Transfer bottle to decapper	0 [0.0, 7.1]	0 [0.0, 7.1]	58 [44.2, 70.6]
L1-17	Transfer open bottle to workbench	36 [24.1, 49.9]	56 [42.3, 68.8]	60 [46.2, 72.4]
L1-18	Transfer plate from rack to base	14 [7.0, 26.2]	22 [12.8, 35.2]	56 [42.3, 68.8]
L1-19	Remove plate lid	92 [81.2, 96.8]	48 [34.8, 61.5]	80 [67.0, 88.8]
L1-20	Grasp pipette and insert tip into liquid	44 [31.2, 57.7]	54 [40.4, 67.0]	72 [58.3, 82.5]
L1-21	Aspirate with second arm	24 [14.3, 37.4]	46 [33.0, 59.6]	32 [20.8, 45.8]
L1-22	Retract pipette and transfer to dispensing position	92 [81.2, 96.8]	100 [92.9, 100.0]	98 [89.5, 99.6]
L1-23	Dispense with second arm and retract	8 [3.2, 18.8]	28 [17.5, 41.7]	20 [11.2, 33.0]
L1-24	Return pipette to rack	8 [3.2, 18.8]	0 [0.0, 7.1]	2 [0.4, 10.5]
L1-25	Grasp spreader and move to agar	56 [42.3, 68.8]	72 [58.3, 82.5]	60 [46.2, 72.4]
L1-26	Spread liquid	70 [56.2, 80.9]	74 [60.4, 84.1]	86 [73.8, 93.0]
L1-27	Discard spreader	46 [33.0, 59.6]	70 [56.2, 80.9]	72 [58.3, 82.5]
L1-28	Place antibiotic disc	2 [0.4, 10.5]	0 [0.0, 7.1]	2 [0.4, 10.5]
L1-29	Close plate lid	0 [0.0, 7.1]	0 [0.0, 7.1]	0 [0.0, 7.1]
L1-30	Open incubator lid	32 [20.8, 45.8]	80 [67.0, 88.8]	8 [3.2, 18.8]

Task	Operation	$\pi_{0.5}$	RDT	π_0
L1-31	Transfer plate into incubator	74 [60.4, 84.1]	18 [9.8, 30.8]	76 [62.6, 85.7]
L1-32	Close incubator lid	8 [3.2, 18.8]	20 [11.2, 33.0]	2 [0.4, 10.5]

Table 14: L2 multi-operation task results: SR with Wilson 95% intervals and ordered progress (PG), all in percent; $N = 50$ episodes per task. Repeated task names denote separate tasks in different workflow contexts.

	$\pi_{0.5}$		RDT		π_0	
Multi-operation task	SR [95% CI]	PG	SR [95% CI]	PG	SR [95% CI]	PG
Workflow A: streak-plate workflow						
L2-01 Mix and open sample bottle	4 [1.1, 13.5]	10.00	0 [0.0, 7.1]	3.33	0 [0.0, 7.1]	2.00
L2-02 Transfer and open plate	22 [12.8, 35.2]	61.00	70 [56.2, 80.9]	85.00	0 [0.0, 7.1]	45.00
L2-03 Streak inoculation	0 [0.0, 7.1]	5.50	14 [7.0, 26.2]	43.00	0 [0.0, 7.1]	7.00
L2-04 Incubation handoff	12 [5.6, 23.8]	42.67	88 [76.2, 94.4]	92.00	0 [0.0, 7.1]	43.33
L2-05 Post-incubation imaging	26 [15.9, 39.6]	63.00	6 [2.1, 16.2]	53.00	2 [0.4, 10.5]	25.00
Workflow B: spread-plate / disc workflow						
L2-06 Mix and open sample bottle	4 [1.1, 13.5]	6.67	0 [0.0, 7.1]	4.00	0 [0.0, 7.1]	0.00
L2-07 Transfer and open plate	12 [5.6, 23.8]	44.00	52 [38.5, 65.2]	68.00	0 [0.0, 7.1]	36.00
L2-08 Aspirate liquid	0 [0.0, 7.1]	5.33	0 [0.0, 7.1]	6.67	0 [0.0, 7.1]	4.00
L2-09 Dispense liquid	2 [0.4, 10.5]	33.00	0 [0.0, 7.1]	41.00	2 [0.4, 10.5]	37.00
L2-10 Spread-plate inoculation	0 [0.0, 7.1]	46.00	2 [0.4, 10.5]	55.33	4 [1.1, 13.5]	44.00
L2-11 Place disc and close plate	0 [0.0, 7.1]	8.00	0 [0.0, 7.1]	9.00	0 [0.0, 7.1]	8.00
L2-12 Incubation handoff	8 [3.2, 18.8]	33.33	66 [52.2, 77.6]	80.00	4 [1.1, 13.5]	51.33

Table 15: Macro-averaged SR (%) with equal weight for each task.

Level	Instances	$\pi_{0.5}$	RDT	π_0
L0	16	97.25	91.75	89.00
L1	32	42.56	43.75	51.81
L2	12	7.50	24.83	1.00

C ADDITIONAL ANALYSES

Table 16: Post-hoc grouping of multi-operation tasks. Entries give mean SR (%) and successes / episodes. Plate handling comprises L2-02/04/07/12; the other eight multi-operation tasks form the second group.

Multi-operation task group	$\pi_{0.5}$	RDT	π_0
Plate handling (4)	13.50 (27/200)	69.00 (138/200)	1.00 (2/200)
Other multi-operation tasks (8)	4.50 (18/400)	2.75 (11/400)	1.00 (4/400)

With 50 trials per task, the within-group task mean equals the pooled success fraction. The mean, median, and counts of tasks at zero or at least 80% SR summarize the distribution of task performance.

Table 17 gives Wilson 95% intervals for the response-study SRs and ordered progress for normal execution. After response training, normal-workflow PG decreases for all loop models and increases only for $\pi_{0.5}$ on spreaders.

Table 17: Loop and spreader results (%), $N = 50$ episodes per model/tool/group at 30k steps. SR includes Wilson 95% intervals. PG is reported for the three-step normal-workflow groups G1 and G3; for the single-step disposal groups G2, G4, and G5, PG equals SR and is omitted. Definitions follow Table 2.

Group / metric	$\pi_{0.5}$	RDT	π_0
(a) Inoculating loop (L1-06/07/08)			
G1 SR	22 [12.8, 35.2]	60 [46.2, 72.4]	10 [4.3, 21.4]
G2 SR	24 [14.3, 37.4]	56 [42.3, 68.8]	18 [9.8, 30.8]
G3 SR	12 [5.6, 23.8]	58 [44.2, 70.6]	0 [0.0, 7.1]
G4 SR	100 [92.9, 100.0]	100 [92.9, 100.0]	100 [92.9, 100.0]
G5 SR	20 [11.2, 33.0]	18 [9.8, 30.8]	0 [0.0, 7.1]
G1 PG	34.00	69.33	17.33
G3 PG	18.67	68.00	2.00
(b) Spreader (L2-10)			
G1 SR	62 [48.2, 74.1]	94 [83.8, 97.9]	0 [0.0, 7.1]
G2 SR	0 [0.0, 7.1]	90 [78.6, 95.7]	2 [0.4, 10.5]
G3 SR	54 [40.4, 67.0]	24 [14.3, 37.4]	0 [0.0, 7.1]
G4 SR	92 [81.2, 96.8]	90 [78.6, 95.7]	86 [73.8, 93.0]
G5 SR	0 [0.0, 7.1]	18 [9.8, 30.8]	0 [0.0, 7.1]
G1 PG	76.00	96.00	22.00
G3 PG	84.67	44.00	10.67

Cross-level checkpoint evaluation. Tables 18 and 19 report the two directions of cross-level evaluation. We reuse the original level-trained checkpoints without further training and evaluate 50 trials per model and target task. L1 checkpoints receive the full L2 multi-operation task instruction, unseen during L1 training, throughout execution; this direction tests instruction generalization together with operation composition. The 14 isolated operations and three multi-operation tasks add $3 \times (14 + 3) \times 50 = 2{,}550$ episodes to the workflow evaluation. SR intervals follow the Wilson calculation in Appendix B.

Table 18 covers the twelve pre-incubation streaking operations (L1-01–12) and two imaging operations (L1-13–14). Across all 14 tasks, switching from L1 to L2 checkpoints improves, decreases, or leaves SR unchanged on 6/6/2 tasks for $\pi_{0.5}$, 6/7/1 for RDT, and 2/11/1 for π_0. Mean SR decreases from 47.71% to 40.43%, 47.14% to 40.86%, and 57.71% to 21.86%, respectively.

Table 18: **Checkpoint comparison on isolated L1 tasks.** Success rates (%) over 50 trials per cell. L1 and L2 denote the checkpoint's training level.

	$\pi_{0.5}$		RDT		π_0	
Task	L1	L2	L1	L2	L1	L2
L1-01	88	26	72	2	68	0
L1-02	6	24	0	0	68	22
L1-03	64	0	78	14	66	0
L1-04	22	80	34	100	66	76
L1-05	68	80	76	54	92	6
L1-06	68	0	54	12	34	2
L1-07	62	92	100	72	86	52
L1-08	74	96	100	0	96	8
L1-09	0	0	0	32	0	0
L1-10	46	0	94	80	12	90
L1-11	90	74	24	42	80	42
L1-12	20	94	22	74	2	0
L1-13	0	0	0	78	58	4
L1-14	60	0	6	12	80	4
Mean (01–14)	47.71	40.43	47.14	40.86	57.71	21.86

Table 19: **L1 checkpoints evaluated on L2 multi-operation tasks.** Each checkpoint receives the full multi-operation task instruction and runs for 50 trials per task. PG and SR are percentages.

	$\pi_{0.5}$		RDT		π_0	
Multi-operation task	PG	SR	PG	SR	PG	SR
L2-02	3	0	34	0	49	0
L2-03	0	0	0	0	0	0
L2-04	0	0	19.3	0	0	0

Isolated-product reference. For a fixed checkpoint, let p_j denote the success probability estimated from the isolated evaluation of operation j. We form the reference $\widehat{\mathrm{SR}}_{\mathrm{prod}} = 100 \prod_j p_j$ for the corresponding multi-operation task. Table 20 compares this reference with observed multi-operation task SR for both checkpoint levels. Its factors measure success under the isolated task's starting states and instruction; multi-operation task execution carries forward the state reached by preceding operations under the multi-operation task instruction.

RDT changes from a product above observed SR to one below observed SR on both L2-02 and L2-04 when switching from L1 to L2 checkpoints. Other comparisons reveal different patterns: $\pi_{0.5}$'s L2 checkpoint has a 64% product and 22% multi-operation task SR on L2-02, while its zero isolated SR on incubator opening accompanies 12% multi-operation task SR on L2-04. RDT likewise obtains zero isolated SR on loop disposal with its L2 checkpoint, yet completes L2-03 at 14%. These comparisons expose the dependence of operation performance on the execution context.

Table 20: **Isolated-product baseline and observed multi-operation task success.** The product uses the same checkpoint's constituent-operation SRs under isolated evaluation. All values are percentages; observed multi-operation task SR uses 50 trials per cell.

		L1 checkpoint		L2 checkpoint	
Multi-operation task	Model	Product	Multi-operation task SR	Product	Multi-operation task SR
L2-02	$\pi_{0.5}$	14.96	0	64.00	22
L2-02	RDT	25.84	0	54.00	70
L2-02	π_0	60.72	0	4.56	0
L2-03	$\pi_{0.5}$	0.00	0	0.00	0
L2-03	RDT	0.00	0	0.00	14
L2-03	π_0	0.00	0	0.00	0
L2-04	$\pi_{0.5}$	8.28	0	0.00	12
L2-04	RDT	4.96	0	24.86	88
L2-04	π_0	0.19	0	0.00	0

Recovering operation-level progress. L2-02 scores plate transfer and lid removal as two ordered milestones, and L2-04 scores incubator opening, plate transfer, and incubator closure as three. Let R_j be the percentage of episodes completing the first j milestones. For these multi-operation tasks,

$$\mathrm{PG} = \frac{1}{K} \sum_{j=1}^{K} R_j, \qquad \mathrm{SR} = R_K. \tag{4}$$

For L2-02 this gives $R_1 = 2\mathrm{PG} - \mathrm{SR}$. With the L1 checkpoints, the reported PG/SR pairs yield transfer completion rates of 6%, 68%, and 98%. The outcome shares are $100 - R_1$ for incomplete transfer, $R_1 - \mathrm{SR}$ for transfer without completed lid removal, and SR for full completion, as shown in Table 5b.

Failure-frequency annotation. Table 6 summarizes 100 failed trajectories per model across L1 and L2 tasks, evaluated with the corresponding level-trained checkpoints. Percentages are computed within each model's 100 analyzed failures. Error accumulation describes an earlier pose or grasp error carried into a later operation; state ambiguity describes repeated or ineffective actions around uncertainty about whether an object is grasped or retained; phase confusion describes execution of

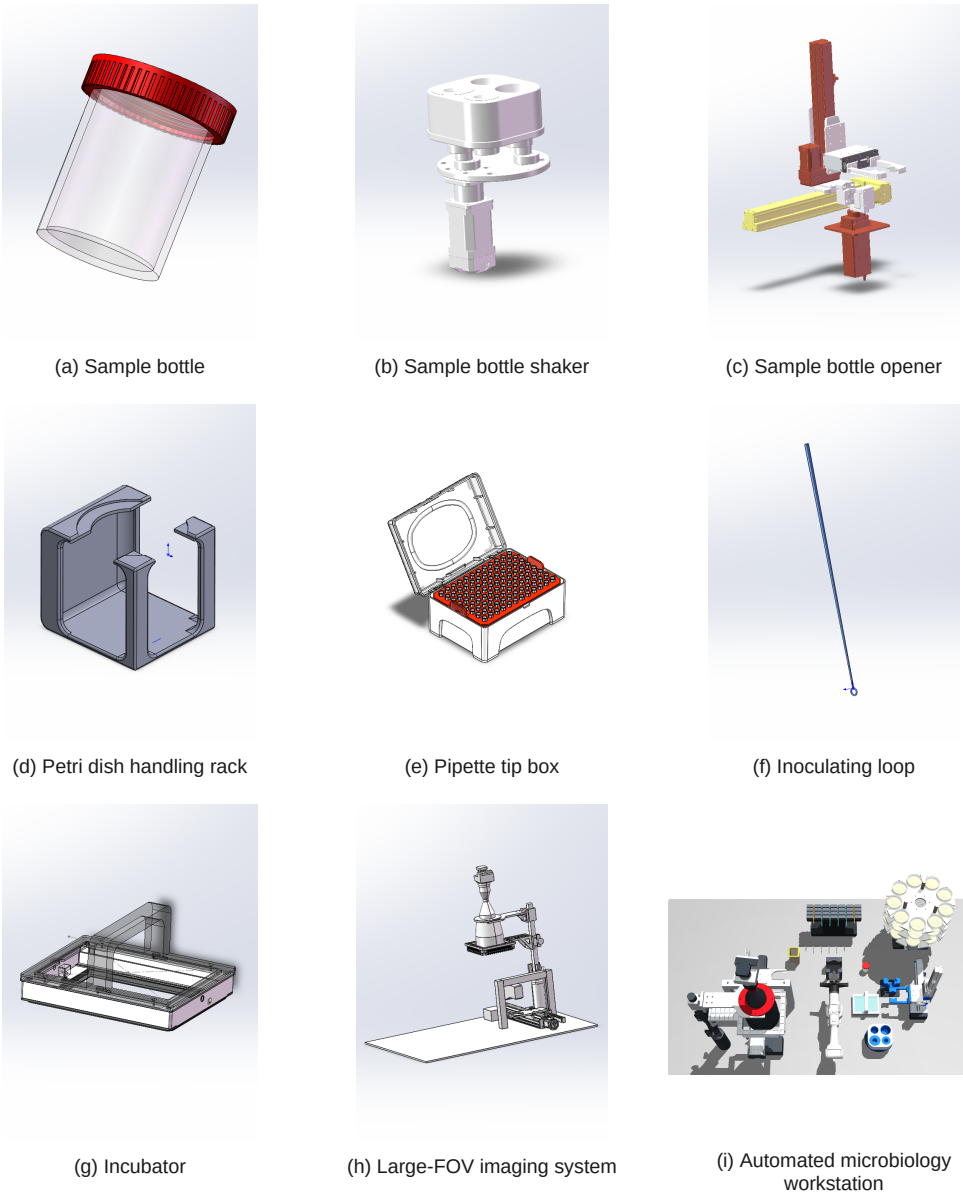
an operation inappropriate to the current workflow stage. Other execution failures comprise the remaining 9%, 6%, and 7% for $\pi_{0.5}$, RDT, and π_0, respectively.

D LABORATORY ASSET INVENTORY

The asset inventory organizes reusable objects and their manipulation interfaces. Figure 14 expands the CAD examples with eight laboratory models, including bottle fixtures, consumable holders, incubation, and imaging hardware. These geometries define the receiving slots, support surfaces, and working ends used by the L1 operations.

Table 21: The 18-entry laboratory asset inventory and associated functional components.

ID	Asset	Type	Movable	Moving parts
A01	Petri dish	Container	Yes	Petri dish lid
A02	Sample bottle	Container	Yes	Bottle cap
A03	Laboratory waste container	Container	Yes	No
A04	Pipette tip box	Container	Yes	Box lid
A05	Pipette tip	Consumable	Yes	No
A06	Cell spreader	Tool	Yes	No
A07	Inoculating loop	Tool	Yes	No
A08	Pipette	Tool	Yes	Button
A09	Inoculation tool holder	Rack	Yes	No
A10	Antibiotic disc holder	Rack	Yes	No
A11	Petri dish handling rack	Rack	Yes	No
A12	Pipette rack	Rack	Yes	No
A13	Piper robotic arm	Robotic arm	No	6-axis joints, gripper
A14	Sample bottle shaker	Instrument	Yes	Shaking tray
A15	Sample bottle opener	Instrument	No	Sliding stage, lifting mechanism, rotary mechanism, gripper
A16	Automated petri dish storage system	Instrument	No	Multilevel movable dish racks
A17	Incubator	Instrument	Yes	Incubator top lid
A18	Large-FOV imaging system	Instrument	No	Sample stage



(a) Sample bottle

(b) Sample bottle shaker

(c) Sample bottle opener

(d) Petri dish handling rack

(e) Pipette tip box

(f) Inoculating loop

(g) Incubator

(h) Large-FOV imaging system

(i) Automated microbiology workstation

Figure 14: **CAD assets and automated microbiology workstation.** (a) The sample bottle defines the body–cap interface. (b, c) The shaker and opener provide bottle-receiving fixtures. (d) The Petri dish handling rack supports plate manipulation. (e) The pipette tip box arranges tip positions beneath a hinged lid. (f) The inoculating loop separates the shaft from the small working end. (g) The incubator provides the plate-placement region and enclosure. (h) The large-field-of-view imaging system positions the plate beneath the camera. (i) The automated microbiology workstation integrates a robotic arm with tool storage, plate storage, and laboratory instruments.

E CONTACT-AWARE EVALUATION PROTOCOL

The loop and spreader study tracks eligibility at the tool's critical working surface using phase-specific contact allowlists. Prohibited contact makes that surface ineligible for the remainder of the response episode. Success then requires disposal without further sample or agar contact.

Instructions and conditions. Groups 1 and 3 use the standard workflow instruction. Groups 2, 4, and 5 append the rule: "If the tool's working end contacts a prohibited surface, stop the normal procedure and discard the tool immediately; do not contact the sample or agar with it again." A script injects contact after loop dipping, before streaking, or after spreader pickup, before its first agar contact; bacterial suspension is deposited by pipette. The script moves the working end to a predefined contamination point, then returns control to the model for its next action. Groups 2 and 4 use four seen sites: the four corners of the plate rack. Group 5 uses locations outside these corners, such as the workbench, within the same tool and protocol phase.

Training and scoring. Policies train on L1-06/07/08 (loop) or L1-25/26/27 (spreader), ending normal and response episodes in disposal. Standard-only and Response-augmented share pretrained initialization and standard demonstrations; the latter adds 100 demonstrations of correct disposal after scripted injection. Both are evaluated at 30k steps. Normal SR scores ordered completion; PG measures the completed prefix. Response SR scores disposal without renewed sample or agar contact. Each model–tool–group combination uses 50 trials, totaling 1,500 (Table 17).

E.1 ILLUSTRATED CONTAMINATION RESPONSES

Figures 15 and 16 show loop and spreader responses to prohibited working-end contact. Each row presents initialization, two intermediate stages, and termination; red circles and linked insets locate the contact and subsequent tool motion. OOD denotes held-out contact locations, as in Group 5.

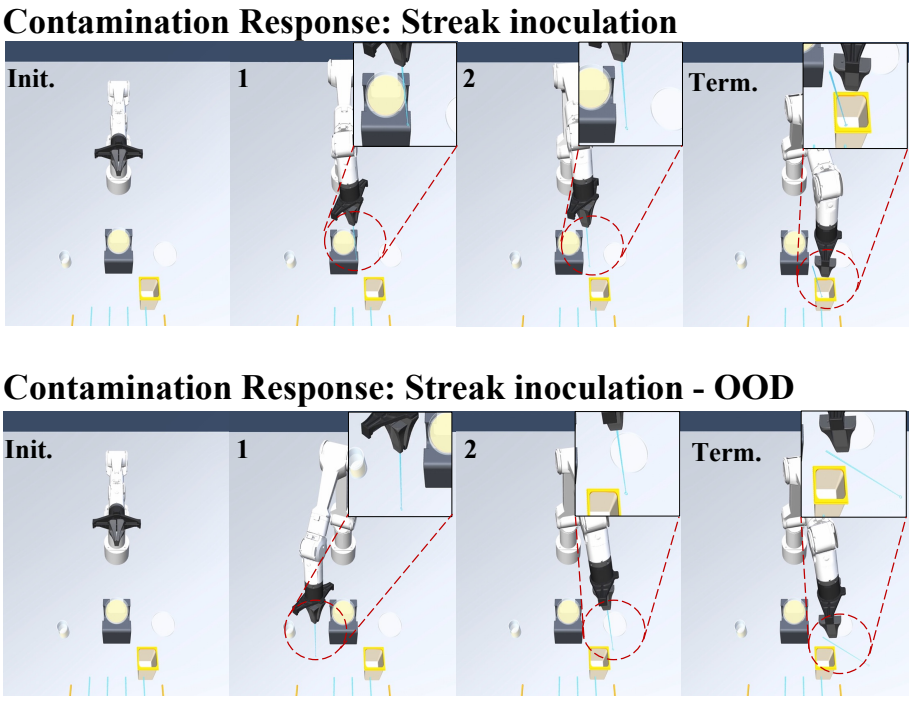


Figure 15: **Inoculating-loop responses to contaminating contact.** Top: the loop's working end contacts the exterior region beside the plate, withdraws from that region, and moves into the waste container. Bottom (OOD): contact occurs on the workbench beside the plate; the robot moves the loop toward the waste container, but the terminal frame shows it released to the right of the opening.

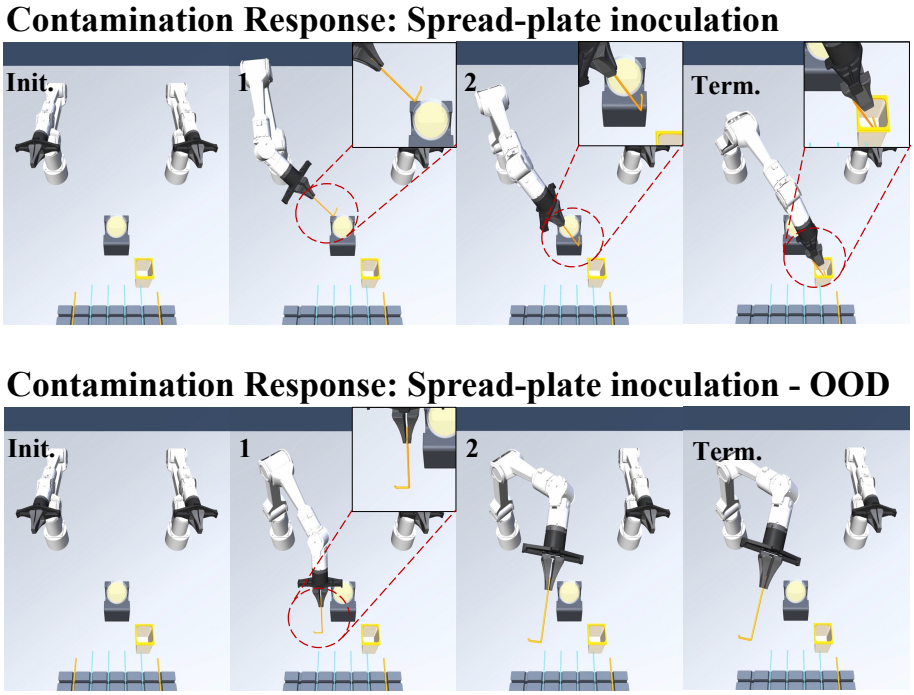


Figure 16: **Spreader responses to contaminating contact.** Top: the spreader's working end contacts the exterior of the plate-support region, moves away from the plate, and enters the waste container. Bottom (OOD): the working end contacts the workbench beside the plate. The robot subsequently holds the spreader away from the agar, with the tool still grasped outside the waste container at termination. This sequence stops tool use but leaves the disposal requirement unmet.

F PHYSICAL TASK EVALUATION

Physical tests use a 6-axis Piper robotic arm. We use NVIDIA A800 for model training and NVIDIA RTX 3090 for inference.

Physical demonstrations cover pipette pickup and immersion (L1-20), lid removal (L1-05/19), streaking (L1-07), and spreader pickup and placement at the agar surface (L1-25). Teleoperation provides 200 trajectories per operation, totaling 800. Tasks were selected for laboratory relevance and their success in simulation.

The hardware comparison in Table 8 evaluates pipette immersion, lid removal, spreader grasping, and streaking with $\pi_{0.5}$, RDT, and π_0. Each model–task pair uses 50 trials and scores completion of the local operation, yielding 600 episodes across the four tasks. Simulation counterparts achieve 44–72%, 48–92%, 56–72%, and 62–100% SR, respectively, across models and workflow contexts. Hardware policies are trained separately for each task for 30k steps on its physical demonstrations.

F.1 PHYSICAL TASK SEQUENCES

Figures 17 and 18 show all four physical operations. Each row presents initialization, two intermediate stages, and termination in the same format as Figure 5. The insets expose the tool grasp, working-end position, and lid contact during manipulation.

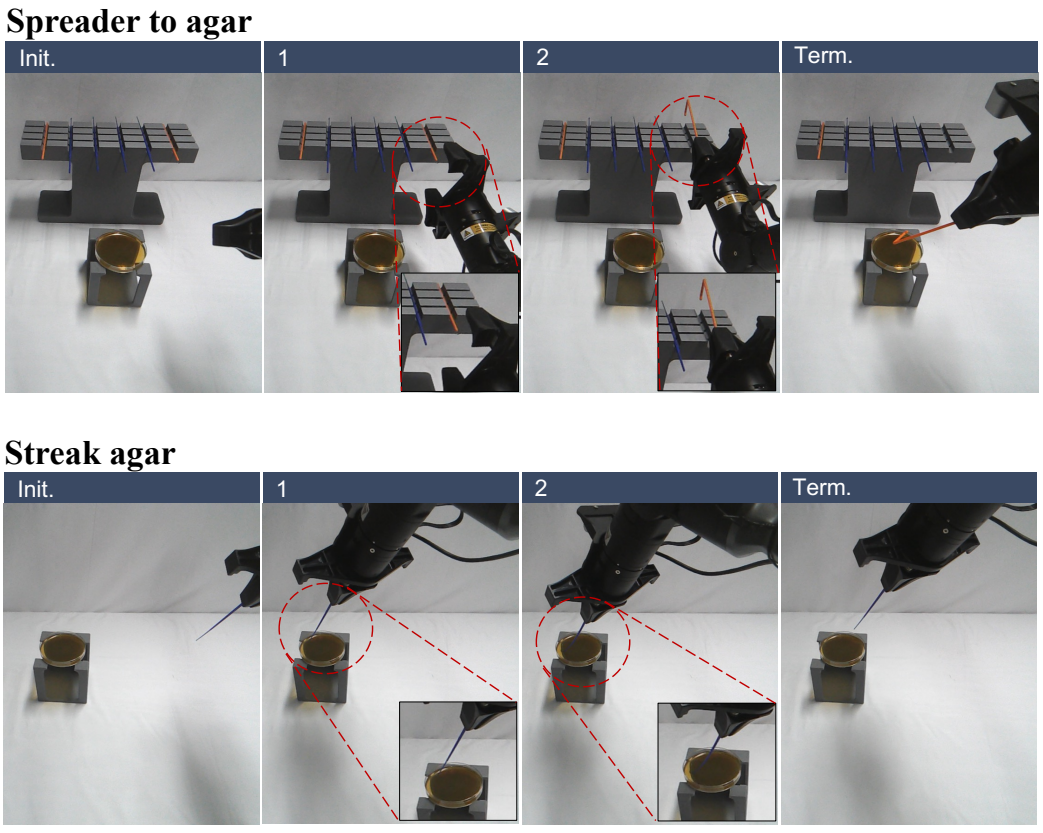


Figure 17: **Physical tool manipulation.** Top: the gripper acquires the spreader from the holder and brings its working end to the agar region (L1-25). Bottom: the held inoculating loop approaches the plate and traverses the agar surface (L1-07). Insets highlight tool acquisition and tool–agar interaction.

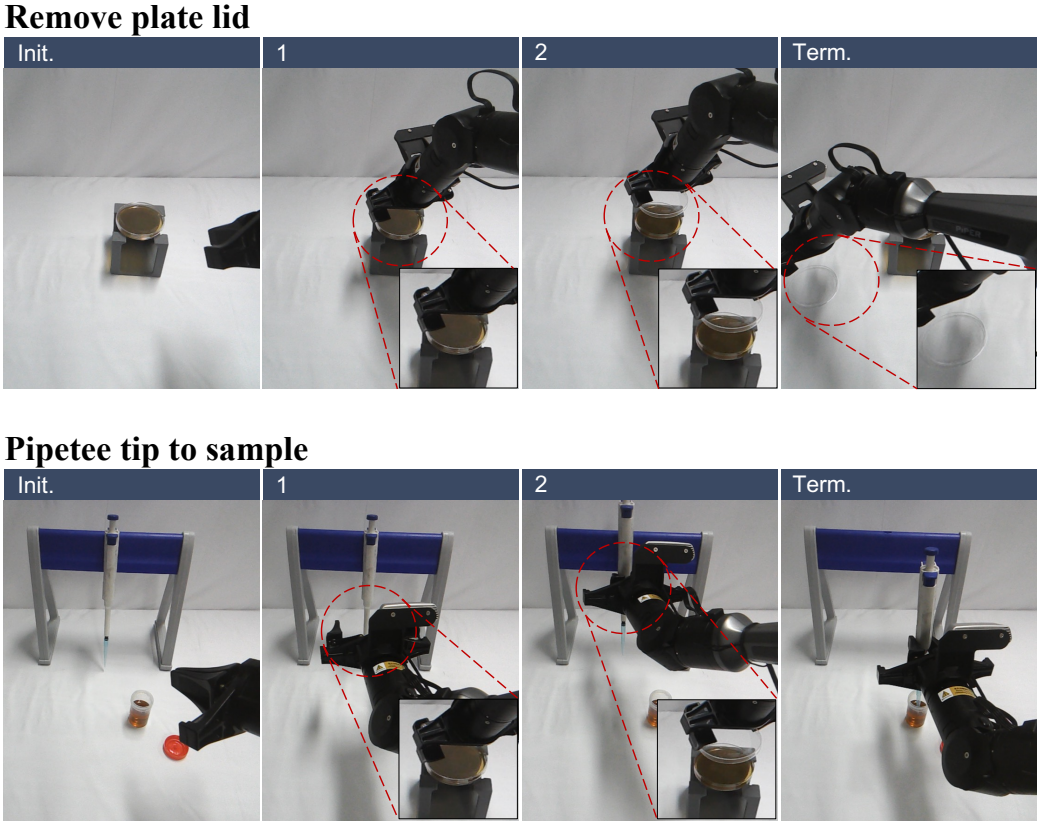


Figure 18: **Physical plate handling and pipette immersion.** Top: the gripper acquires the plate lid, lifts it clear of the plate, and moves it to the staging area (L1-05/19). Bottom: the gripper takes the pipette from the rack and lowers its tip into the sample bottle (L1-20). Insets show lid separation and the pipette grasp and tip position.

G MULTI-AGENT SCENE-SYNTHESIS PROMPTS

All three scene-synthesis agents use Qwen3.7-Flash (Alibaba Cloud, 2026). The system in Section 4 passes a scene brief from the coordination agent to the asset-grounding agent, then passes the grounded instances to the layout agent. The layout agent specifies semantic zones and placement priorities. The deterministic layout stage assigns metric poses, and the MuJoCo and robot-feasibility tools return validation results for revision. The following example instruction and system prompts specify each agent's inputs, responsibilities, and output.

G.1 EXAMPLE USER INSTRUCTION

> Place dual robotic arms at home. Place one closed Petri dish and one sealed sample bottle on the workbench. Place the pipette rack with three pipettes on the front-right side. Place the inoculation tool holder assembly on the front-left side. Place the laboratory waste bin beside the sample bottle.

G.2 SCENE COORDINATION AGENT

> You are the Scene Coordination Agent for a neutral-language scene synthesis system.
>
> Read the user instruction and normalize it into a complete scene brief. Preserve the requested robot configuration, every explicit quantity, object state, spatial zone, object relation, and hard physical requirement. Do not choose local asset IDs, source paths, metric coordinates, or rotations. Do not omit or reinterpret an explicit user constraint.
>
> Write one concise paragraph describing what the scene must contain and what must remain true when it is built. Include the physical checks required for acceptance: compilation, support, non-penetration, workbench bounds, IK, joint limits, and sampled robot paths.

G.3 SEMANTIC ASSET GROUNDING AGENT

> You are the Semantic Asset Grounding Agent for a neutral-language scene synthesis system.
>
> Read the scene brief and the supplied local asset catalog. Resolve every requested object to an existing validated catalog asset and expand quantities into concrete instances. Preserve all states, spatial zones, and relations from the scene brief. Use only catalog-provided asset IDs, source paths, and footprints. If an object cannot be grounded, report the missing match instead of inventing an asset.
>
> Write one concise paragraph naming the grounded instances and their required relations and zones. Do not assign metric coordinates or rotations; those are owned by the deterministic layout stage.

G.4 CONSTRAINT-AWARE EMBODIED LAYOUT AGENT

> You are the Constraint-Aware Embodied Layout Agent for a neutral-language scene synthesis system.
>
> Read the scene brief, the grounded asset instances, their footprints, and any validation feedback. Convert the requested spatial intent into a semantic layout plan. Preserve assembly relations, robot-base clearance, reachable interaction space, workbench bounds, support, and non-penetration. If validation feedback is present, revise only the affected placement intent.
>
> Write one concise paragraph assigning semantic zones and ordering priorities for the deterministic layout tool. Do not create assets and do not output metric coordinates or rotations. The deterministic layout and MuJoCo tools compile, place, simulate, validate, and accept or reject the scene.

H TRAINING AND INFERENCE DETAILS

Camera views. The single-arm inoculating-loop scene uses two camera views: `overview` and `wrist`. The dual-arm spreader scene uses three views: `overview`, `right_wrist`, and `left_wrist`.

Observation inputs. π_0 and $\pi_{0.5}$ receive images from the current timestep, the current robot state, and the language instruction. RDT receives a two-frame image history, the current robot state, and the language instruction. Visual inputs use the camera views specified above for each scene.

Learning rates and batch size. All models use a global batch size of 32. RDT uses a learning rate of 1×10^{-4}. For real-world tasks, RDT continues training from the corresponding L1 checkpoint with a learning rate of 5×10^{-5}.

For π_0 and $\pi_{0.5}$, the learning rate increases from a small initial value during the first 1,000 training steps, reaching a peak of 2.5×10^{-5} at the end of warmup. It then follows a cosine decay schedule, reaching 2.5×10^{-6} at the end of training, one tenth of the peak rate.

Action prediction. We set the action chunk lengths with reference to RoboDojo (Chen et al., 2026). For π_0 and $\pi_{0.5}$, both the action chunk length and the action horizon are 25 steps. RDT uses an action chunk length of 64 steps.