LIBERO-INDUSTRY: BENCHMARKING ROBOTIC MANIPULATION UNDER INDUSTRIAL PROCESS CONSTRAINTS

Anonymous authors
Paper under double-blind review

ABSTRACT

Industrial manipulation in flexible manufacturing, e.g., high-mix, low-volume (HMLV) electronics assembly, requires robot policies to adapt to changing products and production conditions. Such settings impose industrial process constraints, including product variants, moving material flows, finite execution windows, and tight geometric tolerances. However, existing evaluation frameworks do not systematically capture these constraints when assessing generalist robot policies. To address this gap, we introduce LIBERO-INDUSTRY, a novel benchmark built on the widely used robot learning benchmark LIBERO, that translates the aforementioned industrial constraints into controllable task conditions and measurable success criteria. Specifically, LIBERO-INDUSTRY comprises three classic task scenarios: Static-to-Dynamic (S2D) loading, Dynamic-to-Static (D2S) unloading, and Dynamic-to-Dynamic (D2D) transfer. The suites contain 216 tasks in total that systematically vary object geometry, conveyor speed, interaction-window duration, and geometric acceptance region size. Extensive experiments across five generalist robot policies (including vision-language-action models and world-action models) show that performance progressive degrades as industrial constraints are introduced and combine, with benchmark code and datasets to be made publicly available. Real-world experiments demonstrate that LIBERO-INDUSTRY substantially improves physical task success over standard LIBERO under matched training conditions. Code and data will be made publicly available.

1 INTRODUCTION

Flexible manufacturing, e.g., high-mix, low-volume (HMLV) electronic assembly, requires frequent adaptation to changing products and production conditions, requiring substantial re-engineering effort on conventional automation (Gan et al., 2023; Schlenoff et al., 2021). Generalist robot policies, including Vision-Language-Action (VLA) models such as RT-2 (Brohan et al., 2023) and Octo (Ghosh et al., 2024), offer a learned alternative by conditioning robot actions on multimodal observations and language instructions across diverse tasks. However, industrial readiness requires more than task-level adaptability, particularly under the precision and operating conditions encountered in industrial settings (Li et al., 2025).

Industrial manipulation in flexible production is shaped by industrial constraints, introducing product variants, moving material flows, finite execution windows, and tight geometric tolerances, as illustrated in Figure 1. Product variants may require different setups and processing times (Gan et al., 2023), making frequent production changes harder to accommodate with fixed systems (Koren et al., 1999). Moving material flows impose finite time windows for manipulation (Islam et al., 2021), while parts must meet tight geometric tolerances despite manufacturing and perception errors(Jiang et al., 2022). Traditional automation works reliably when tasks and environments remain stable, but frequent changes often require reprogramming, recalibration, robot re-teaching, or changes to the workcell (Schlenoff et al., 2021; Gan et al., 2023) (detailed in Appendix 4). Together, these characteristics define the industrial process constraints that embodied manipulation evaluations should capture.

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.

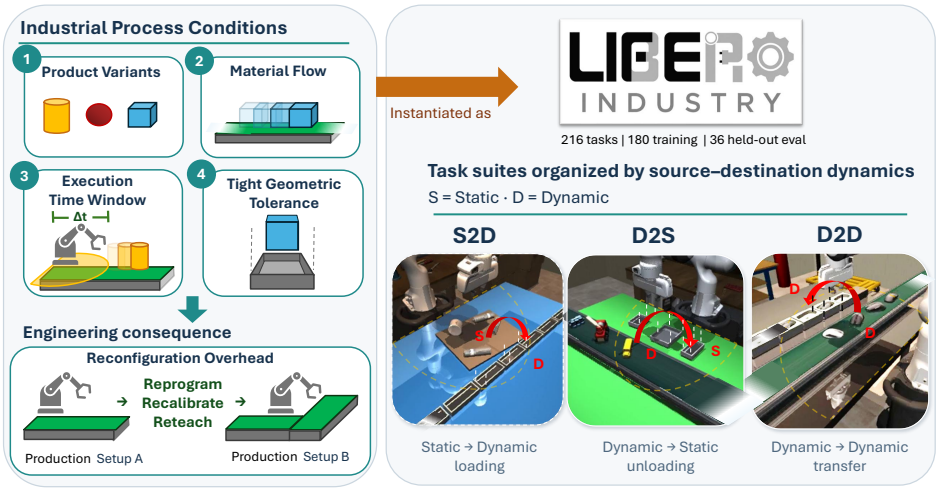


Figure 1: **LIBERO-INDUSTRY overview.** Motivated by industrial process constraints, the benchmark evaluates generalist robot policies across three classic industrial scenarios.

However, existing embodied-manipulation and manufacturing-oriented benchmarks do not systematically evaluate generalist robot policy performance under industrial constraints, as summarised in Table 2. On one hand, general-purpose manipulation benchmarks, such as LIBERO, CALVIN, and Meta-World, provide standardized protocols for studying task learning, generalization, and language-conditioned control (Liu et al., 2023; Mees et al., 2022; Yu et al., 2019), yet largely abstract away the timing, motion, and placement requirements imposed by production processes. On the other hand, manufacturing-oriented research and benchmarks, including the Agile Robotics for Industrial Automation Competition (ARIAC), address robotic flexibility, dynamic production conditions, and operational performance (Harrison et al., 2018; Jiang et al., 2022), do not comprehensively evaluate these industrial constraints or generalist robot policies. Consequently, how moving sources and destinations, finite execution windows, and tight geometric tolerances affect task success, throughput, and stage-wise failures remains underexplored, particularly as these requirements accumulate across multi-object workloads.

To address this research gap, we introduce LIBERO-INDUSTRY, an extension of the widely used robot learning benchmark LIBERO (Liu et al., 2023) for evaluating generalist robot policies under industrial process constraints, that operationalizes these constraints as controllable task conditions and measurable success criterion. As shown in Figure 1, LIBERO-INDUSTRY comprises three task suites: Static-to-Dynamic (S2D) loading, Dynamic-to-Static (D2S) unloading and Dynamic-to-Dynamic (D2D) transfer, that are derived from typical industrial scenarios. vary whether workpiece sources and destinations are stationary or moving, capturing different forms of material-flow interaction. Each suite includes three progressive difficulty levels with controlled variations in workspace geometry, visual conditions, and task configuration, together with structured perturbations informed by industrial operating conditions and standards. Tasks couple spatial completion requirements with finite execution windows, allowing process constraints to be introduced systematically and in combination.

We evaluate five generalist robot policies, including VLA models and world-action model (WAM) in simulation, measuring task success, throughput, and stage-wise completion. We further assess real-world performance through physical D2S trials on an AUBO-i5H arm, comparing fine-tuning effectiveness with LIBERO-INDUSTRY and LIBERO under matched conditions.

In summary, our main contributions are as follows:

(a) **Industrial Process Benchmark:** We introduce LIBERO-INDUSTRY, a novel benchmark that systematically evaluates product variants, moving material flows, finite execution windows, and tight geometric tolerances.

Table 1: Comparison of recent tabletop manipulation simulation benchmarks across embodied-AI and industrial-process evaluation dimensions. $\sim$ denotes partial coverage; for Product variants, object-level variation alone does not establish a dedicated production-variant evaluation.

Dataset	Domain	Product variants	Material flow	Time window	Geom. tolerance	Real-world validation
General-purpose embodied-AI benchmarks						
LIBERO	Household	$\sim$	✗	✗	✗	✗
LIBERO-PLUS	Household	$\sim$	✗	✗	✗	✗
VLA-Arena	Household	$\sim$	✗	✗	✗	✗
LIBERO-X	Household	$\sim$	✗	✗	✗	✗
Dynamic VLA	Generalised	$\sim$	✗	✗	✗	✓
DOMINO	Generalised	$\sim$	✗	✗	✗	✓
Manufacturing-oriented simulation benchmarks						
ARIAC 2025	EV battery	✓	✓	✗	✗	✗
AI-CPS	Industry	$\sim$	✗	✗	✗	✗
RoboBPP	Logistics	✓	✓	✗	$\sim$	✗
LIBERO-Industry **(ours)**	Industry	✓	✓	✓	✓	✓

(b) **Process-Constrained Task Design:** We translate industrial process requirements as controllable task conditions, including conveyor velocity, bounded execution duration, and geometric completion requirement sizes, enabling their systematic evaluation individually and in combination.

(c) **Empirical Evaluation and Failure Diagnosis:** We benchmark five generalist policies, including VLA models and WAM, showing progressive performance degradation as industrial constraints are introduced and combined, and demonstrate considerable improvements in real-world task success over standard LIBERO.

2 Related Work

2.1 Generalist Robot Policies

Generalist policies map visual observations and language instructions to actions across tasks. RT-2 and OpenVLA use tokenized action generation (Brohan et al., 2023; Kim et al., 2024), whereas π_0 and $\pi_{0.5}$ model continuous actions with generative objectives (Black et al., 2024; Physical Intelligence, 2025b). LIBERO-Industry evaluates adapted policies from these families under common process constraints.

2.2 Embodied Manipulation Benchmarks

Early robot-manipulation benchmarks established standardized evaluation for multi-task learning and long-horizon control. Meta-World evaluates multi-task and meta-reinforcement learning, CALVIN targets language-conditioned long-horizon execution, and LIBERO measures knowledge transfer across sequentially learned task suites (Yu et al., 2019; Mees et al., 2022; Liu et al., 2023). Recent VLA benchmarks shift attention toward robustness: LIBERO-PRO, LIBERO-PLUS, VLA-Arena, and LIBERO-X evaluate sensitivity to visual, physical, spatial, or linguistic variations (Zhou et al., 2025; Fei et al., 2025; Zhang et al., 2025; Wang et al., 2026).

Dynamic-manipulation benchmarks complement these efforts by introducing moving objects or environmental disturbances. DOMINO evaluates manipulation under dynamic environmental changes,

while DynamicVLA targets closed-loop control for dynamic objects (Fang et al., 2026; Xie et al., 2026). Together, these benchmarks broaden evaluation beyond fixed task configurations, but they do not jointly parameterize production flow, finite physical opportunity windows, task-specific placement conditions, and multi-object workloads.

2.3 Industrial Process-Aware Evaluation

Manufacturing-oriented evaluation efforts provide complementary test settings: ARIAC assesses simulated production workflows (National Institute of Standards and Technology, 2025), AI-CPS evaluates controllers on industrial manipulation tasks in Isaac Sim (Zhou et al., 2024), and RoboBPP benchmarks robotic online bin packing with physics-based simulation and industrial datasets (Wang et al., 2025). Industrial robotics evaluates a manipulator not only as a task solver but also as a component of a production system. HMLV manufacturing requires flexibility under frequent product and workcell changes, while retaining reliable execution and process coordination (Gan et al., 2023; Huettemann et al., 2016; Johansen et al., 2021). Research on robotic assembly and line balancing accordingly emphasizes planning, execution, workstation coordination, and production performance (Jiang et al., 2022; Fathi et al., 2024).

Production-system literature distinguishes continuous and intermittent material transport, while takt and cycle-time concepts relate execution to operating schedules and demand (Groover, 2014; Linck & Cochran, 1999; Mönch et al., 2021). Robot standards such as ISO 9283 provide established terminology for pose accuracy and repeatability (International Organization for Standardization, 1998). However, these studies typically evaluate dedicated systems, robot properties, or production configurations rather than language-conditioned generalist policies across standardized task distributions. Conversely, embodied benchmarks emphasize generalization but rarely expose industrial process conditions as controlled variables.

3 LIBERO-Industry Benchmark Design

The benchmark aims to evaluate how reliably and efficiently generalist robot policies perform material-transfer tasks in changing industrial workcells. To support comparable evaluation, each task instance is assigned explicit completion predicates and a predeclared episode horizon derived from its transport schedule.

3.1 Preliminaries and Task Interface

We evaluate policies with different architectures and training objectives through a common language-conditioned observation–action interface, following LIBERO and CALVIN (Liu et al., 2023; Mees et al., 2022):

$$o_t \in \mathcal{O}, \qquad \mathbf{u}_t \sim \pi_\theta(\cdot \mid o_{\le t}, \ell_m), \qquad \mathbf{u}_t \in \mathcal{U}, \tag{1}$$

where ℓ_m is the instruction for evaluation episode m, o_t contains the benchmark-defined visual and proprioceptive observations, and $\mathbf{u}_t$ is a robot action produced at control frequency f_c. The full simulator state s_t is available to the evaluator for computing task predicates but need not be available to the policy.

We denote the task instance for evaluation episode m by $\mathcal{T}_m$, with $\mathcal{C}_m$ collecting its temporal opportunities and goal specification:

$$\begin{aligned} \mathcal{T}_m &= \left\langle \ell_m, \mu_{0,m}, \mathcal{W}_m, \xi_m^{\mathrm{src}}, \xi_m^{\mathrm{dst}}, \mathcal{C}_m \right\rangle, \\ \mathcal{C}_m &= \left\langle \{(a_{mj}, H_{mj})\}_{j=1}^{N_{\mathrm{target},m}}, \mathcal{G}_m \right\rangle. \end{aligned} \tag{2}$$

Here, $\mu_{0,m}$ is the initial-state specification; $\mathcal{W}_m$ contains the target workpieces and any pre-existing destination workpieces; and ξ_m^{src} and ξ_m^{dst} denote the *source process* and *destination process*, respectively, each of which may be static, continuously moving, or intermittent. For each of the $N_{\mathrm{target},m}$ target workpieces, indexed by j, a_{mj} is its availability time and H_{mj} is the end of its physical opportunity horizon. The set $\mathcal{G}_m$ contains the goal propositions that define valid completion for the target workpieces and, when present, pre-existing destination workpieces.

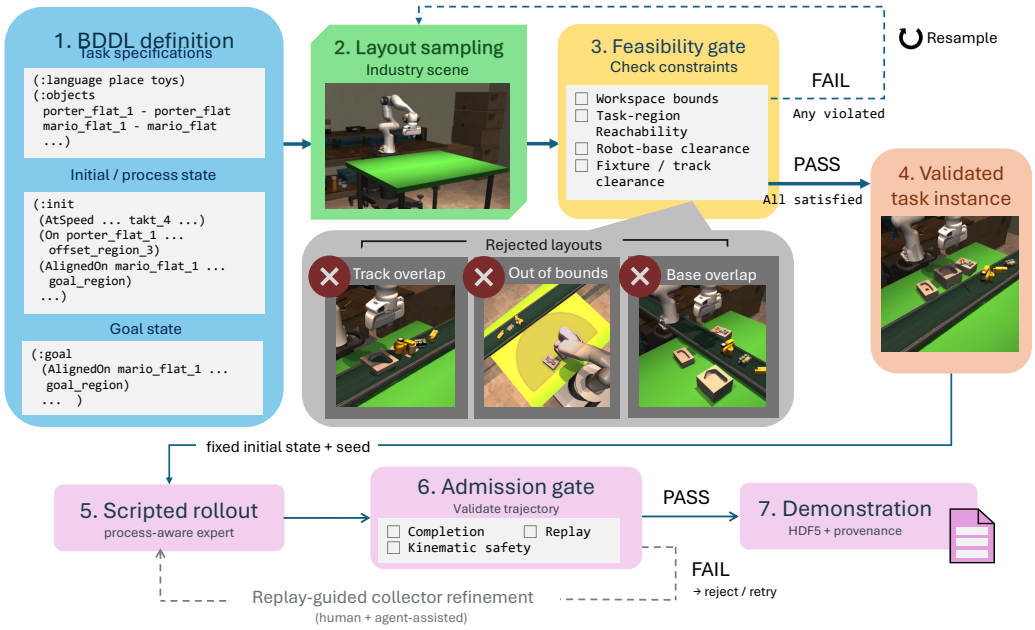


Figure 2: **Task generation and scripted demonstration collection in LIBERO-INDUSTRY.** BDDL specifies objects, process states, and goals; sampled layouts pass a feasibility gate or are resampled (top). A task-specific scripted expert attempts a rollout from a fixed initial state and seed (bottom). The diagram shows completion, kinematic-safety, and replay checks before accepted trajectories are stored as HDF5 demonstrations with provenance. Failed attempts can prompt replay-guided collector refinement.

3.2 INDUSTRIAL PROCESS CONSTRAINTS

Each task has two process-level conditions: transport determines how the source and destination evolve, while temporal opportunity determines when a transfer remains feasible.

Transport condition. Under a continuous source process, target workpieces move at signed velocity V_s; under an intermittent destination process, the destination moves at signed velocity V_d between stationary service periods. These transport states change the workpiece or goal geometry over time. Speed is therefore an operating variable rather than a success metric. The continuous and intermittent modes follow established material-transport terminology (Groover, 2014).

Temporal opportunity condition. For the continuous source process, target workpieces separated by centre-to-centre pitch d_{src} have transport-event interval

$$\tau_{\text{src}} = \frac{d_{\text{src}}}{|V_s|}. \tag{3}$$

For the pulse-indexed destination process, slot pitch s_{slot} and stationary service dwell T_{dwell} define the transport-event interval

$$\tau_{\text{dst}} = T_{\text{dwell}} + \frac{s_{\text{slot}}}{|V_d|}. \tag{4}$$

These intervals define the physical transport schedule, while $[a_{mj}, H_{mj}]$ bounds the opportunity during which workpiece j can be completed. For the selected source arrival intervals, see Appendix B.4.

3.3 PROCESS-AWARE PROCEDURAL TASK GENERATION

Figure 2 illustrates the task generation pipeline . While workspace support follows LIBERO's region-based formulation (Liu et al., 2023), our generator additionally conditions object initialization and task timing on material-flow dynamics. While workspace support follows LIBERO's

Table 2: Task taxonomy and dataset composition of LIBERO-INDUSTRY across the three workflow suites. Full combinations are defined over object, conveyor speed, motion direction, and difficulty level. For D2S, four objects are selected as a stratified subset from six available objects.

Suite	Workflow logic	Full combos	Instantiated	Train	Test
		Obj. × Speed × Dir. × Lvl.			
D2S	Conveyor Belt → Precision Slot	$4 \times 3 \times 2 \times 3$	72	60	12
S2D	Material Plate → Dynamic Slot	$4 \times 3 \times 2 \times 3$	72	60	12
D2D	Conveyor Belt → Dynamic Slot	$4 \times 3 \times 2 \times 3$	72	60	12
Total			216	180	36

region-based formulation (Liu et al., 2023), our generator conditions object initialization and task timing on material-flow dynamics. Given an operation anchor, static fixtures are sampled with level-specific pose perturbations; moving sources are back-projected using transport direction, velocity, arrival interval, and offset; moving destinations set slot spacing from geometry and travel direction and service intervals from velocity and dwell (Appendix B.2). Each layout is serialized as BDDL with its initial predicates, transport conditions, instruction, and goals, and infeasible layouts are resampled before instantiation.

3.4 TASK SUITES AND DIFFICULTY LEVELS

The source and destination transport states produce three non-static transfer suites, summarized in Table 2.

- **Dynamic-to-Static (D2S) unloading**: moving source → stationary fixture. The policy must acquire a continuously transported workpiece before its source opportunity ends and then satisfy the fixture-specific completion predicate.
- **Static-to-Dynamic (S2D) loading**: stationary source → intermittently moving destination. The policy must place a workpiece during an available destination service period and maintain the required placement as the carrier resumes motion.
- **Dynamic-to-Dynamic (D2D) transfer**: moving source → intermittently moving destination. The policy must satisfy both the source-acquisition opportunity and the destination-service opportunity before stable placement can be credited.

The benchmark instantiates these suites with workpieces from multiple industrial domains and level-specific destination fixtures (for asset inventory see Appendix B.1).

For task instance m, the benchmark generator converts the instantiated transport schedule and completion allowance into a rollout budget:

$$T_{\text{roll},m} = t_{\text{start},m} + T_{\text{event},m} + T_{\text{complete},m} + t_{b,m}, \qquad K_{\text{roll},m} = \lceil f_c T_{\text{roll},m} \rceil , \tag{5}$$

where $t_{\text{start},m}$ is the initialization or first-arrival offset, $T_{\text{event},m}$ is the time spanned by the relevant transport events, $T_{\text{complete},m}$ is the final manipulation or service allowance, and $t_{b,m}$ is the terminal buffer. Appendix B.4 expands this expression for D2S, S2D, and D2D and reports the fixed allowances for every suite, level, and transport condition. These values are shared by all evaluated policies and are benchmark time limits rather than universal factory takt times.

Each level preserves the same high-level transfer objective while jointly increasing completion precision and environmental variation: Level 1 pairs a broad tray region with the nominal workcell condition, Level 2 adds object-matched grooves with prescribed orientations and bounded perturbations, and Level 3 combines form-fitting inlays with broader perturbations. For moving destinations, success additionally requires carrier retention during transport, with perturbations specified using the pose-accuracy and pose-repeatability terminology of ISO 9283 (International Organization for Standardization, 1998) (see Secs. B.5.3, B.5.2, and B.6).

3.5 Evaluation Protocol and Metrics

We report binary task success as the primary metric and two Partial Completion Rates as complementary metrics.

Primary metric: success rate (SR). For each $g \in \mathcal{G}_m$, let $E_m(g) \in \{0, 1\}$ denote its Boolean evaluation at the predeclared rollout boundary $K_{\text{roll},m}$. Episode success and Task Success Rate over M evaluation episodes are

$$Y_m = \prod_{g \in \mathcal{G}_m} E_m(g), \qquad \text{SR} = \frac{1}{M} \sum_{m=1}^{M} Y_m. \tag{6}$$

Complementary metrics: partial completion rate (PCR). Partial Completion Rates measure progress toward the transfer objective. For episode m, let $N_{\text{transferred},m}$ denote the number of target workpieces credited by the transfer predicate, $N_{\text{base},m}$ the number of pre-existing destination workpieces, and $N_{\text{slotted},m}$ the number of target and pre-existing workpieces satisfying their required terminal predicates. The episode-level scores are

$$\text{PCR}_{\text{T},m} = \frac{N_{\text{transferred},m}}{N_{\text{target},m}}, \qquad \text{PCR}_{\text{S},m} = \frac{N_{\text{slotted},m}}{N_{\text{target},m} + N_{\text{base},m}}. \tag{7}$$

The first score isolates progress on target transfers, whereas the second also penalizes displacement of pre-existing destination workpieces. We report their episode means:

$$\text{PCR}_{\text{T}} = \frac{1}{M} \sum_{m=1}^{M} \text{PCR}_{\text{T},m}, \qquad \text{PCR}_{\text{S}} = \frac{1}{M} \sum_{m=1}^{M} \text{PCR}_{\text{S},m}. \tag{8}$$

4 Experiments

This section presents the experimental evaluation of LIBERO-Industry. We first introduce the experimental setup and evaluated policies (Section 4.1), followed by simulation evaluation (Section 4.2), diagnostic analyses, and physical validation on D2S manipulation (Section 4.4).

Table 3: **Evaluation-set results.** Scores in %; 360 rollouts per policy. PCR_{S} includes initially valid destination workpieces that remain valid.

Policy	SR	PCR_{T}	PCR_{S}
$\pi_{0.5}$	0.0	0.4	12.5
OpenVLA-OFT	0.0	0.0	11.2
UniVLA	0.0	0.0	12.2
X-VLA	0.0	0.0	11.9
Cosmos Policy	0.0	0.0	11.8

4.1 Experimental Setup

We follow the original LIBERO simulation settings except that each rollout is bounded by the task-specific horizon $K_{\text{roll},m}$ (Eq. equation 5). We evaluate five publicly available LIBERO-finetuned checkpoints: $\pi_{0.5}$, OpenVLA-OFT, UniVLA, X-VLA, and Cosmos Policy. The deterministic evaluation set contains 36 tasks: 12 each from S2D, D2S, and D2D, with four tasks per difficulty level in each suite; with 10 initial states per task. We report SR, PCR_{T}, and PCR_{S} following Section 3.5 (details in Appendix C.1).

4.2 Benchmark Results

Across the 36 held-out tasks, none of the five checkpoints completes a full rollout: SR is 0.0% (0/360) for every policy (Table 3). The suite-level breakdown likewise shows zero SR in S2D, D2S,

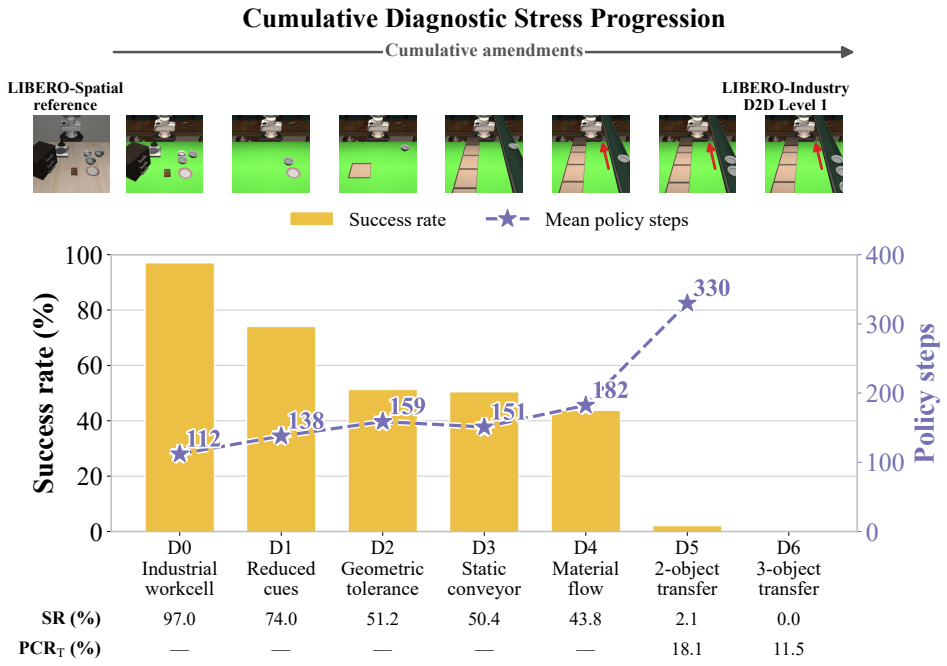


Figure 3: **Cumulative diagnostic stress progression for $\pi_{0.5}$.** Each bar denotes complete-task success of the stress settings for n trials (D0–D1: $n = 200$; D2–D6: $n = 240$), and $\mathrm{PCR_T}$ denotes mean target-object transfer completion. Stars show mean policy steps over successful trajectories; D6 has no successful trajectories. The progression is diagnostic rather than causal.

and D2D (Appendix C.2). Target-transfer completion is nearly absent. Only $\pi_{0.5}$ records nonzero $\mathrm{PCR_T}$: 1.1% in S2D and 0.4% overall. Its D2S and D2D scores, and all three suite scores for the other four policies, are 0.0%. These transfer credits do not satisfy the full task-success predicate. Overall $\mathrm{PCR_S}$ ranges from 11.2% to 12.5%. This metric also credits pre-existing destination workpieces that remain valid (Section 3.5); a nonzero score therefore need not indicate a new target transfer. The small differences in $\mathrm{PCR_S}$ should be interpreted as complementary placement-predicate outcomes, not as differences in complete-task success.

4.3 Ablation and Findings

The D0–D6 diagnostic progression evaluates $\pi_{0.5}$ on 20 trials per task under cumulative scene, object, and conveyor shifts. SR is 97.0% at D0, drops to roughly 50% over D2–D4 (51.2%, 50.4%, and 43.8%), reflecting an extent of generalization of placement target from the original LIBERO dataset, and then collapses at D5–D6 (2.1% and 0.0%). The sharpest degradation occurs when object count increases from one to two objects (D4 to D5), where mean policy steps rise to 330, indicating that additional objects and longer per-object pick-and-place sequences drive the failure. Partial transfer remains without full task completion ($\mathrm{PCR_T} = 18.1\%$ at D5 and 11.5% at D6), with reach-stage failures accounting for 132/240 D5 and 202/240 D6 episodes (full ablation tracker and representative failure trajectories are provided in Appendix C.3 and Appendix C.3.3).

4.4 Real-World Experiments

Experimental setting. We evaluate two sim–real co-training setups initialized from the same $\pi_{0.5}$ base checkpoint: **S1:** 1,724 original-LIBERO simulation dataset + 95 real-world D2S task demonstrations; and **S2:** 1,412 LIBERO-Industry simulation dataset + 95 real-world D2S task demonstrations. Both use a 9:1 simulation-to-real mini-batch sampling ratio, trained for 30k step, and evaluated with the same protocol on the Aubo–CTAG D2S workcell. Thus, the comparison isolates the simulation-data source while holding the base checkpoint, physical data, and co-training protocol fixed (Appendix E).

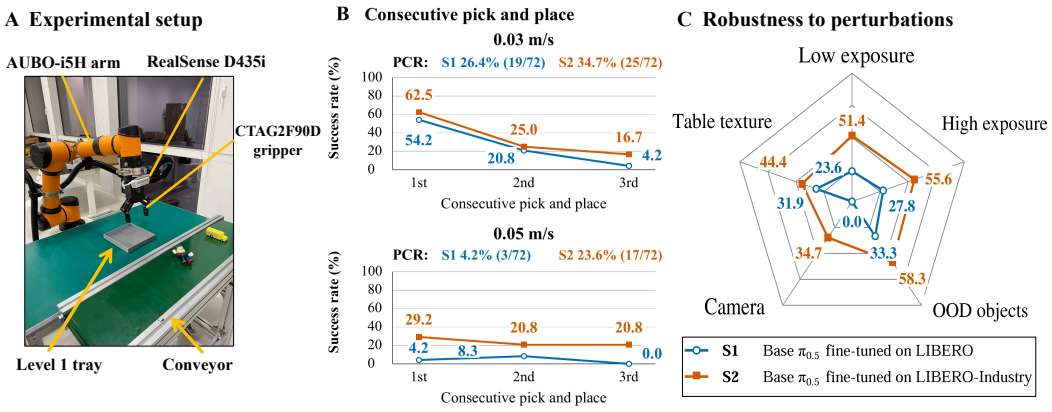


Figure 4: **Real-world validation. A:** Aubo–CTAG D2S workcell. **B:** Per-step release-on-tray success for **S1** and **S2** at conveyor speeds of 0.03 and 0.05 m/s over 24 trials per condition; the table reports aggregate target-transfer completion as $\mathrm{PCR_T}$ out of 72 planned placements. **C:** Performance under five physical perturbations, with exact SR labeled on the curves.

Comparison across co-training setups. Figure 4B compares the two resulting policies under dynamic-to-static unloading at two conveyor speeds. At 0.03 m/s, **S2** attains a completion rate of 34.7% (25/72), compared with 26.4% (19/72) for **S1**. At 0.05 m/s, the corresponding rates are 23.6% (17/72) for **S2** and 4.2% (3/72) for **S1**. The difference between the two setups is therefore more pronounced at the higher conveyor speed, where the available interaction time is shorter.

Performance across consecutive placements shows a similar distinction. At 0.03 m/s, the third-placement SR is 16.7% for **S2** and 4.2% for **S1**; at 0.05 m/s, **S2** retains a 20.8% SR on the third placement, whereas **S1** records 0.0%. These results show that the policy co-trained with LIBERO-INDUSTRY simulation data maintains higher completion rates than that with original-LIBERO simulation data, with the largest separation observed at the faster conveyor speed.

Generalization under operating-condition variations. Figure 4C compares the two co-training setups under changes in illumination, object appearance, camera configuration, and table texture. **S2** attains higher measured SR than **S1** across all five evaluated variations, including 51.4% versus 23.6% under low exposure, 55.6% versus 27.8% under high exposure, 58.3% versus 33.3% with out-of-distribution objects, 34.7% versus 0.0% under the camera variation, and 44.4% versus 31.9% with the altered table texture. Thus, the performance difference between the two co-training setups is also observed beyond the nominal D2S configuration across the tested physical operating-condition changes.

5 CONCLUSION AND FUTURE DIRECTIONS

We introduced LIBERO-INDUSTRY, a benchmark for evaluating generalist policies under industrial process constraints, including product variability, material flows, finite execution-time windows, and tight geometric tolerance. Built on LIBERO, the benchmark organizes these conditions into S2D , D2S and D2D task suites, with controlled variations in task configuration and operating conditions. A key future direction is to determine whether generalist policies can retain flexibility without sacrificing industrial production rates; the benchmark's finite opportunity windows abstract this requirement from heterogeneous factory needs. Future evaluation should move toward production-calibrated settings that relate cycle times to empirical operating rates and test repeated material arrivals, recovery events, and long-horizon throughput. Ultimately, industrial readiness should be assessed not only by task completion but by compatibility with diverse production processes.

AI USE STATEMENT

We used generative AI tools as interactive software-engineering and writing assistants. During the development of task-specific scripted controllers for simulation demonstration generation, these tools assisted with inspecting source code, diagnostic logs, and replay-derived evidence; proposing and implementing localized controller revisions; and creating or updating focused tests. They also assisted with literature discovery, manuscript organization, drafting and language editing. Human authors defined the task semantics and protected collection inputs, selected the diagnostic evidence, reviewed proposed revisions and validation results, and retained responsibility for all experimental decisions. At collection time, trajectory admission was determined by predeclared task-completion, data-integrity, and replay-verification criteria rather than by generative AI. Generative AI also assisted with refining benchmark and evaluation procedures and interpreting diagnostic results; the authors checked these suggestions against the implementation and experimental records. The authors reviewed all AI-assisted code and manuscript content and take full responsibility for the final methods, data, results, claims, and text. Because no human-only comparator was conducted, we make no claim that the use of generative AI tools improved development efficiency.

ETHICS STATEMENT

LIBERO-INDUSTRY extends LIBERO with procedurally generated simulation data and robot demonstrations collected in a physical workcell. The study evaluates robot policies rather than human participants, although a teleoperator appears in the collection photographs. Before publicly sharing photographs or recorded camera views, we will confirm permission to publish identifiable people and screen the data for incidental identifying content. The benchmark also uses third-party 3D assets. Public releases of code, data, and modified assets will include source and license information and will exclude materials for which redistribution rights have not been confirmed. The reported physical results are research evaluations and do not establish that the policies are safe for unsupervised industrial deployment.

REPRODUCIBILITY STATEMENT

Section 3 defines the benchmark tasks, generalization levels, evaluation protocol, and metrics. Section 4 reports the simulation comparisons and diagnostic analyses. The appendix documents the asset inventory (Appendix B.1), timing and episode horizons (Appendix B.4), evaluation procedure (Appendix 3.5), and reported model configurations (Appendix C.1).

REFERENCES

Kevin Black, Noah Brown, Danny Driess, Adnan Esmail, Michael Equi, Chelsea Finn, Niccolo Fusai, Lachy Groom, Karol Hausman, Brian Ichter, et al. π_0: A vision-language-action flow model for general robot control. *arXiv preprint arXiv:2410.24164*, 2024. doi: 10.48550/arXiv.2410.24164. URL https://arxiv.org/abs/2410.24164.

Anthony Brohan, Noah Brown, Justice Carbajal, Yevgen Chebotar, Xi Chen, Krzysztof Choromanski, Tianli Ding, Danny Welker, Zhenghua Xu, Fei Xia, et al. RT-2: Vision-language-action models transfer web knowledge to robotic control. *arXiv preprint arXiv:2307.15818*, 2023.

Cheng Chi, Siyuan Feng, Yilun Du, Zhenjia Xu, Eric Cousineau, Benjamin Burchfiel, and Shuran Song. Diffusion policy: Visuomotor policy learning via action diffusion. In *Robotics: Science and Systems (RSS)*, 2023.

Laura Downs, Anthony Francis, Nate Koenig, Brandon Kinman, Ryan Hickman, Krista Reymann, Thomas B. McHugh, and Vincent Vanhoucke. Google scanned objects: A high-quality dataset of 3d scanned household items. In *IEEE International Conference on Robotics and Automation (ICRA)*, pp. 2553–2560, 2022. doi: 10.1109/ICRA46639.2022.9811809.

Heng Fang, Shangru Li, Shuhan Wang, Xuanyang Xi, Dingkang Liang, and Xiang Bai. Towards generalizable robotic manipulation in dynamic environments. *arXiv*, 2026. doi: 10.48550/arXiv.2603.15620.

Masood Fathi, Arash Sepehri, Morteza Ghobakhloo, Mohammad Iranmanesh, and Ming-Lang Tseng. Balancing assembly lines with industrial and collaborative robots: Current trends and future research directions. *Computers & Industrial Engineering*, 193:110254, 2024. doi: 10.1016/j.cie.2024.110254.

Senyu Fei, Siyin Wang, Junhao Shi, Zihao Dai, Jikun Cai, Pengfang Qian, Li Ji, Xinzhe He, Shiduo Zhang, Zhaoye Fei, Jinlan Fu, Jingjing Gong, and Xipeng Qiu. Libero-plus: In-depth robustness analysis of vision-language-action models. *arXiv*, 2025. doi: 10.48550/arXiv.2510.13626.

Zhi Lon Gan, Siti Nurmaya Musa, and Hwa Jen Yap. A review of the high-mix, low-volume manufacturing industry. *Applied Sciences*, 13(3):1687, 2023. doi: 10.3390/app13031687.

Caelan Reed Garrett, Rohan Chitnis, Rachel Holladay, Beomjoon Kim, Tom Silver, Leslie Pack Kaelbling, and Tomás Lozano-Pérez. Integrated task and motion planning. *Annual Review of Control, Robotics, and Autonomous Systems*, 4(1):265–293, 2021.

Dibya Ghosh, Homer Rich Walke, Karl Pertsch, Kevin Black, Oier Mees, Sudeep Dasari, Joey Hejna, Tobias Kreiman, Charles Xu, Jianlan Luo, You Liang Tan, Lawrence Yunliang Chen, Quan Vuong, Ted Xiao, Pannag R Sanketi, Dorsa Sadigh, Chelsea Finn, and Sergey Levine. Octo: An open-source generalist robot policy. In *Proceedings of Robotics: Science and Systems*, 2024. doi: 10.15607/RSS.2024.XX.090.

Mikell P. Groover. *Automation, Production Systems, and Computer-Integrated Manufacturing*. Pearson, Upper Saddle River, NJ, 4th edition, 2014.

Tuomas Haarnoja, Aurick Zhou, Pieter Abbeel, and Sergey Levine. Soft actor-critic: Off-policy maximum entropy deep reinforcement learning with a stochastic actor. In *International Conference on Machine Learning (ICML)*, pp. 1861–1870, 2018.

William Harrison, Anthony Downs, and Craig I. Schlenoff. The agile robotics for industrial automation competition. *AI Magazine*, 39(4), 2018. doi: 10.1609/aimag.v39i4.2795.

Neville Hogan. Impedance control: An approach to manipulation: Part i–theory. *Journal of Dynamic Systems, Measurement, and Control*, 107(1):1–7, 1985.

Guido Huettemann, Christian Gaffry, and Robert H. Schmitt. Adaptation of reconfigurable manufacturing systems for industrial assembly—review of flexibility paradigms, concepts, and outlook. *Procedia CIRP*, 52:112–117, 2016. doi: 10.1016/j.procir.2016.07.021.

International Organization for Standardization. ISO 9283:1998 manipulating industrial robots — performance criteria and related test methods. Standard, International Organization for Standardization, Geneva, Switzerland, 1998. URL `https://www.iso.org/standard/22244.html`.

International Organization for Standardization. ISO 8995-1:2002 lighting of work places — part 1: Indoor. Standard, International Organization for Standardization, Geneva, Switzerland, 2002. URL `https://www.iso.org/standard/28857.html`.

Fahad Islam, Oren Salzman, Aditya Agarwal, and Maxim Likhachev. Provably constant-time planning and replanning for real-time grasping objects off a conveyor belt. *The International Journal of Robotics Research*, 2021. doi: 10.1177/02783649211027194.

Yuze Jiang, Zhouzhou Huang, Bin Yang, and Wenyu Yang. A review of robotic assembly strategies for the full operation procedure: Planning, execution and evaluation. *Robotics and Computer-Integrated Manufacturing*, 78:102366, 2022. doi: 10.1016/j.rcim.2022.102366.

Kerstin Johansen, Sagar Rao, and Milad Ashourpour. The role of automation in complexities of high-mix in low-volume production—a literature review. *Procedia CIRP*, 104:1452–1457, 2021. doi: 10.1016/j.procir.2021.11.245.

Moo Jin Kim, Karl Pertsch, Siddharth Karamcheti, Ted Xiao, Ashwin Balakrishna, Suraj Nair, Rafael Rafailov, Ethan Foster, Grace Lam, Pannag Sanketi, et al. OpenVLA: An open-source vision-language-action model. *arXiv preprint arXiv:2406.09246*, 2024.

Y. Koren, U. Heisel, F. Jovane, T. Moriwaki, G. Pritschow, G. Ulsoy, and H. Van Brussel. Reconfigurable manufacturing systems. *CIRP Annals*, 48(2):527–540, 1999. doi: 10.1016/S0007-8506(07)63232-6.

Shuai Li, Yizhe Chen, Dong Li, Sichao Liu, Dapeng Lan, Yu Liu, and Zhibo Pang. Transferring vision-language-action models to industry applications: Architectures, performance, and challenges. *arXiv preprint arXiv:2509.23121*, 2025. doi: 10.48550/arXiv.2509.23121.

Joachim Linck and David S. Cochran. The importance of takt time in manufacturing system design. Technical Report 1999-01-1635, SAE International, 1999.

Bo Liu, Yifeng Zhu, Chongkai Gao, Yihao Feng, Qiang Liu, Yuke Zhu, and Peter Stone. LIBERO: Benchmarking knowledge transfer for lifelong robot learning. *arXiv preprint arXiv:2306.03310*, 2023.

Lovano. Conveyor belt assembly line 3d model, 2023. URL https://www.cgtrader.com/free-3d-models/industrial/industrial-machine/conveyor-belt-assembly-line.

Ajay Mandlekar, Soroush Nasiriany, Bowen Wen, Iretiayo Akinola, Yashraj Narang, Linxi Fan, Yuke Zhu, and Dieter Fox. MimicGen: A data generation system for scalable robot learning using human demonstrations. In *Proceedings of the 7th Conference on Robot Learning*, volume 229 of *Proceedings of Machine Learning Research*, pp. 1820–1864. PMLR, 2023. URL https://proceedings.mlr.press/v229/mandlekar23a.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*, 7(3):7327–7334, 2022. doi: 10.1109/LRA.2022.3180108.

Tobias Mönch, Arnd Huchzermeier, and Peter Bebersdorf. Variable takt times in mixed-model assembly line balancing with random customisation. *International Journal of Production Research*, 59(15):4670–4689, 2021. doi: 10.1080/00207543.2020.1769874.

National Institute of Standards and Technology. ARIAC 2025 documentation, 2025. URL https://pages.nist.gov/ARIAC_docs/en/stable/index.html.

Physical Intelligence. Openpi: Open-source models and packages for robotics. https://github.com/Physical-Intelligence/openpi, 2025a. Accessed 2026-09-10.

Physical Intelligence. $\pi_{0.5}$: A vision-language-action model with open-world generalization. *arXiv preprint arXiv:2504.16054*, 2025b. doi: 10.48550/arXiv.2504.16054. URL https://arxiv.org/abs/2504.16054.

ProductViz3D. Skincare cosmetics pack 3d model, 2025. URL https://www.cgtrader.com/free-3d-models/industrial/other/skincare-cosmetics-pack.

Marc H Raibert and John J Craig. Hybrid position/force control of manipulators. *Journal of Dynamic Systems, Measurement, and Control*, 102(2):126–133, 1981.

Craig I. Schlenoff, Zeid Kootbally, and Erez Karpas. Agile robotics for industrial applications: Editorial. *Robotics and Computer-Integrated Manufacturing*, 71:102162, 2021. doi: 10.1016/j.rcim.2021.102162.

Standardization Administration of China. GB/T 10595-2017 belt conveyors. Standard, General Administration of Quality Supervision, Inspection and Quarantine of the People's Republic of China, Beijing, China, 2017. URL https://openstd.samr.gov.cn/bzgk/std/newGbInfo?hcno=699B0A98206696B50EE902FE6AB8BDB4.

Guodong Wang et al. Libero-x: Robustness litmus for vision-language-action models. *arXiv*, 2026. doi: 10.48550/arXiv.2602.06556.

Zhoufeng Wang, Hang Zhao, Juzhan Xu, Shishun Zhang, Ruizhen Hu, Chenyang Zhu, Zecui Zeng, Weiyan Zhu, Zeyu Xiong, Haibin Yu, and Kai Xu. RoboBPP: Benchmarking robotic online bin packing with physics-based simulation, 2025. URL https://arxiv.org/abs/2512.04415.

Daniel E Whitney. Force feedback control of manipulator fine motions. *Journal of Dynamic Systems, Measurement, and Control*, 99(2):91–97, 1977.

Haozhe Xie, Beichen Wen, Jiarui Zheng, Zhaoxi Chen, Fangzhou Hong, Haiwen Diao, and Ziwei Liu. Dynamicvla: A vision-language-action model for dynamic object manipulation. *arXiv*, 2026. doi: 10.48550/arXiv.2601.22153.

Seonghyeon Ye, Yunhao Ge, Kaiyuan Zheng, Shenyuan Gao, Sihyun Yu, George Kurian, Suneel Indupuru, You Liang Tan, Chuning Zhu, Jiannan Xiang, Ayaan Malik, Kyungmin Lee, William Liang, Nadun Ranawaka, Jiasheng Gu, Yinzhen Xu, Guanzhi Wang, Fengyuan Hu, Avnish Narayan, Johan Bjorck, Jing Wang, Gwanghyun Kim, Dantong Niu, Ruijie Zheng, Yuqi Xie, Jimmy Wu, Qi Wang, Ryan Julian, Danfei Xu, Yilun Du, Yevgen Chebotar, Scott Reed, Jan Kautz, Yuke Zhu, Linxi Fan, and Joel Jang. World action models are zero-shot policies. *arXiv preprint arXiv:2602.15922*, 2026. doi: 10.48550/arXiv.2602.15922.

Tianhe Yu, Deirdre Quillen, Zhanpeng He, Ryan Julian, Avnish Narayan, Hayden Shively, Adithya Bellathur, Karol Hausman, Chelsea Finn, and Sergey Levine. Meta-World: A benchmark and evaluation for multi-task and meta reinforcement learning. *arXiv*, 2019. doi: 10.48550/arxiv.1910.10897.

Kevin Zakka. obj2mjcf: A utility for converting composite obj files to a mujoco model. `https://github.com/kevinzakka/obj2mjcf`, 2022.

Borong Zhang, Jiahao Li, Jiachen Shen, Yuhao Zhang, Yishuai Cai, Yuanpei Chen, Juntao Dai, Jiaming Ji, and Yaodong Yang. Vla-arena: An open-source framework for benchmarking vision-language-action models. *arXiv*, 2025. doi: 10.48550/arXiv.2512.22539.

Tony Z Zhao, Vikash Kumar, Sergey Levine, and Chelsea Finn. Learning fine-grained bimanual manipulation with low-cost hardware. In *Robotics: Science and Systems (RSS)*, 2023.

Xueyang Zhou, Yangming Xu, Guiyao Tie, Yongchao Chen, Guowen Zhang, Duanfeng Chu, Pan Zhou, and Lichao Sun. Libero-pro: Towards robust and fair evaluation of vision-language-action models beyond memorization. *arXiv*, 2025. doi: 10.48550/arXiv.2510.03827.

Zhehua Zhou, Jiayang Song, Xuan Xie, Zhan Shu, Lei Ma, Dikai Liu, Jianxiong Yin, and Simon See. Towards building AI-CPS with NVIDIA Isaac Sim: An industrial benchmark and case study for robotics manipulation. In *2024 IEEE/ACM 46th International Conference on Software Engineering: Software Engineering in Practice (ICSE-SEIP)*, pp. 263–274, 2024. doi: 10.1145/3639477.3639740.

LIBERO-industry: Benchmarking Robotic Manipulation under Industrial Process Constraints

Appendix

Contents

A Control Paradigms and Technical Comparison **15**

B Benchmark Design and Task Specification **16**

B.1 Task Suites, Assets, and Fixtures 16

B.2 Catalog Composition and Spatial Distribution 19

B.3 Language Instruction Schema 19

B.4 Temporal Protocol and Operating Conditions 20

B.5 Conveyor Mechanics and Success Predicates 22

B.6 Environmental Generalization Settings 24

C Evaluation Protocol and Extended Results **25**

C.1 Evaluated Models and Checkpoints 25

C.2 Evaluation-Set Breakdown 26

C.3 Cumulative Diagnostic Stress Progression 26

D Cross-Embodiment Conversion and Policy Adaptation **28**

D.1 Trajectory Conversion and Recovery 29

E Physical System, Real-World Dataset, and Sim–Real Evaluation **30**

E.1 Physical Workcell and Sensor Setup 30

E.2 Real-World Collection, Splits, and Quality Control 30

E.3 Physical Demonstration Collection and Recorded Views 31

E.4 Sim–Real Co-Training Configuration 33

A CONTROL PARADIGMS AND TECHNICAL COMPARISON

This section provides a structured comparison of traditional, learning-based, and generalist robotic control paradigms. Table 4 outlines the trade-offs in model scale, data dependencies, and physical precision limits that motivate the transition toward high-fidelity simulation environments like LIBERO-INDUSTRY.

Table 4: Comparative Analysis of Control Paradigms in Industrial Manipulation Tasks.

Paradigm	**Representative Arch. / Algos**	**Model Size & Freq.**	**Data / Training Source**	**Precision & Repeatability**
Traditional Rule-Driven	FSM / BT; A* / TAMP; PID, Force, Hybrid Position–Force, Impedance	No learned params; 1000–2000 Hz	Zero data; physical models & expert kinematics	± 0.01 mm positioning accuracy
Modular Perception-Action	YOLOv8 / FastSAM + LLM-TAMP (MoveIt / OMPL)	Perception: 100M–1B+ (< 10 Hz); Planner: Det. (> 200 Hz)	Zero downstream trajectories; pre-trained vision & 3D CAD models	0.5–1.0 mm positioning alignment
Imitation Learning (IL)	ACT, Diffusion Policy, DP3	50M–100M params; 30–100 Hz inference	Behavioral cloning from 50–100 expert demonstrations	Sub-millimeter level
Model-Free / Residual RL	PPO, SAC, OEC-IRRL, RL-100	< 20M params; > 500 Hz	Online/Offline RL via environment reward signals	0.1 mm precision alignment
Vision-Language-Action (VLA)	OpenVLA, RT-2, Octo, RDT-1B	1B–7B+ params; 3–10 Hz	Autoregressive next-token prediction on internet-scale robot data	~ 2.2 cm position error, 12.4° angular error
World Action Models (WAM)	DreamZero, 1XWM, UniSim	14B+ params; 3–7 Hz closed-loop control	Joint video-action diffusion / latent generative dynamics	Coarse visual trajectory tracking (> 1 cm)

As detailed in Table 4, the landscape of robotic manipulation control can be broadly categorized into distinct evolutionary tiers, each exhibiting specific deployment bottlenecks when transitioned to High-Mix Low-Volume (HMLV) manufacturing:

1. **Classical Deterministic Systems:** Classical stacks separate task sequencing, motion planning, and interaction control. Finite-state machines and behavior trees sequence operations, while task-and-motion planners compute feasible, collision-free motions (Garrett et al., 2021). A geometric path $\mathbf{q}(s)$ specifies configuration-space motion, whereas its time-parameterized trajectory $\mathbf{q}(t)$ also accounts for velocity, acceleration, and execution deadlines; conveyor tasks may require online replanning as object-pose estimates change (Islam et al., 2021). *Direct force control* regulates a measured contact wrench, while *hybrid position/force control* assigns motion and force objectives to complementary task-space directions, as in lateral alignment with axial insertion force (Whitney, 1977; Raibert & Craig, 1981). *Impedance control* instead shapes the compliant relation $\mathbf{M}_d\ddot{\mathbf{e}}+\mathbf{D}_d\dot{\mathbf{e}}+\mathbf{K}_d\mathbf{e} = \mathbf{F}_{\text{ext}}$, allowing the end effector to yield during uncertain contact such as groove or inlay insertion (Hogan, 1985). These methods are conceptual baselines rather than evaluated controllers in LIBERO-INDUSTRY: the reported experiments use OSC-POSE at 20 Hz and score release, contact, alignment, and velocity synchronization rather than force-tracking error.

2. **Task-Specific Learning Policies:** Imitation Learning (IL) architectures (Zhao et al., 2023; Chi et al., 2023) and model-free Reinforcement Learning (RL) (Haarnoja et al., 2018) bridge the gap to handling unstructured sensory inputs. While maintaining high-frequency execution suitable for tight-tolerance insertion, IL policies exhibit severe spatial overfitting to fixed demonstration setups and fail under out-of-distribution (OOD) part variants. Conversely, model-free RL achieves high compliance but remains bounded by extreme sample inefficiency during physical training.
3. **Generalist Foundation Models:** Vision-Language-Action (VLA) architectures (Brohan et al., 2023; Kim et al., 2024) and generative World Action Models (WAMs) (Ye et al., 2026) exhibit unprecedented semantic reasoning and few-shot adaptation. However, VLAs suffer from low inference rates (< 10 Hz) and centimeter-scale positioning errors (~ 2.2 cm), making them incapable of fine-grained assembly. Meanwhile, WAMs generate future visual dynamics alongside action streams; while effective for visual forecasting, visual dynamics pre-training alone does not guarantee physical contact compliance or real-world reward execution, highlighting the critical need for standardized, closed-loop simulation benchmarks to rigorously evaluate these models prior to physical line deployment.

B Benchmark Design and Task Specification

B.1 Task Suites, Assets, and Fixtures

Each instantiated task is encoded as a BDDL specification containing its objects, initial conditions, language instruction, and goal propositions. The propositions in $\mathcal{G}_m$ are grounded by the corresponding benchmark predicates.

Figure 5 visualizes the suite–level matrix with one representative task from each combination.

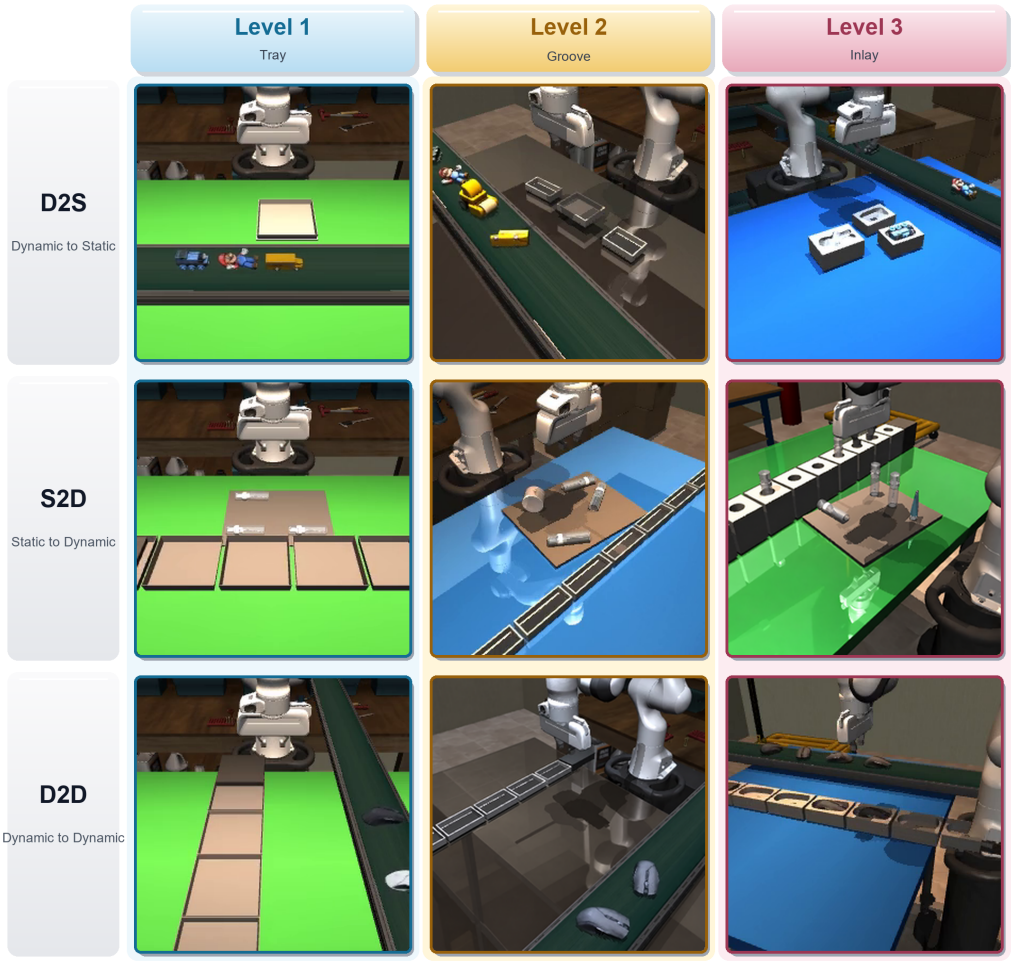


Figure 5: Representative LIBERO-INDUSTRY task instantiations. Rows show Dynamic-to-Static (D2S), Static-to-Dynamic (S2D), and Dynamic-to-Dynamic (D2D) transport. Columns progress from Level 1 sector trays to Level 2 object-specific grooves and Level 3 form-fitting inlays. Header and border colors identify difficulty levels and do not encode measured performance.

We introduce a novel industrial object assets spanning across different manufacturing domains. Base workpiece models are sourced and amended from open-access repositories, including industial conveyor Lovano (2023), Google Scanned Objects Downs et al. (2022) and commercial design configurations ProductViz3D (2025). Table 5 catalogs the validated workpiece inventory and their matching destination fixtures.

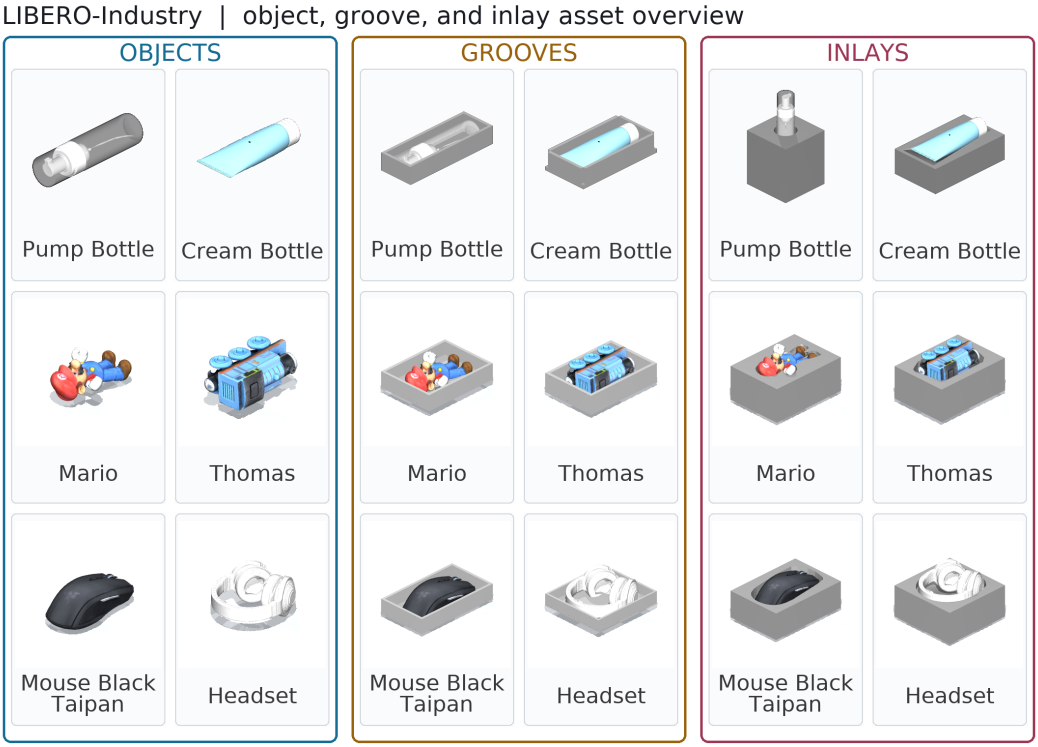


Figure 6: Representative LIBERO-INDUSTRY workpiece and fixture assets. From left to right, the panels show six evaluation workpieces and their object-specific Level 2 grooves and Level 3 form-fitting inlays, spanning personal care, model figurines and vehicles, and consumer-electronics domains.

We enforce a strict **1 cm geometric clearance** rule across three difficulty tiers:

- **Sector Trays (Level 1):** Domain-wide general receptacles dimensioned to fit the maximum bounding box of the largest workpiece within that industrial category plus a 1 cm margin.
- **Grooves (Level 2):** Object-specific slots generated strictly from the bounding box dimensions of individual workpieces plus a 1 cm margin.
- **Form-Fitting Inlays (Level 3):** Object-specific negative-space profiles modeled via proportional 1cm boolean mesh expansion in Blender.

We process inlay configurations through the `obj2mjcf` utility using Volumetric Hierarchical Approximate Convex Decomposition (V-HACD) to decompose hollow geometries into convex submeshes Zakka (2022).

Table 5: Systematic Breakdown of Novel Evaluation Workpieces and Target Mating Fixtures Ordered by Insertion Difficulty

Industrial Domain	Base Workpiece Asset ID	Complementary Fixtures
Workspace	`conveyor_belt`, `conveyor_ghost`	
Model Figurines & Vehicles	`mario`, `grimlock`, `bumblebee`, `dragon`, `mini_fire_engine`, `mini_roller`, `school_bus`, `thomas`, `porter`	`toy_tray`, `*_flat_groove`, `*_upright_groove`, `*_v_groove`, `*_flat_inlay`
Personal Care & Cosmetics	`cream_bottle`, `cream_jar`, `pump_bottle`, `serum_bottle`	`cosmetics_tray`, `*_flat_groove`, `*_upright_groove`, `*_flat_v_groove`, `*_v_groove`, `*_flat_inlay`, `*_upright_inlay`
Consumer Electronics	`mouse_naga`, `mouse_white_taipan`, `mouse_black_taipan`, `headset`, `keyboard`	`electronics_tray`, `*_upright_groove`, `*_v_groove`, `*_upright_inlay`

* Denotes an object-specific fixture where $\star$ maps to the corresponding workpiece Asset ID.

B.2 Catalog Composition and Spatial Distribution

The BDDL catalog specifies target workpieces, non-target objects, and placement regions for each suite and level. Figure 7 compares the centers of its declared placement regions with LIBERO-Spatial; each point represents a region declaration, not a task or recorded demonstration.

Target and confounder composition.
D2D. Level 1 contains four targets without confounders. Level 2 samples two or three targets from four source objects, leaving one or two source confounders. Level 3 contains two or three targets and three confounders distributed across source and destination locations.

D2S. Level 1 contains three moving targets. Level 2 contains three targets and one source confounder. Level 3 contains two moving targets, one or two source confounders, and one prealigned destination object.

S2D. Level 1 contains targets without confounders. Level 2 contains three targets with one or two source confounders. Level 3 contains three targets with upright/flat pose variation and up to three source and destination confounders in total.

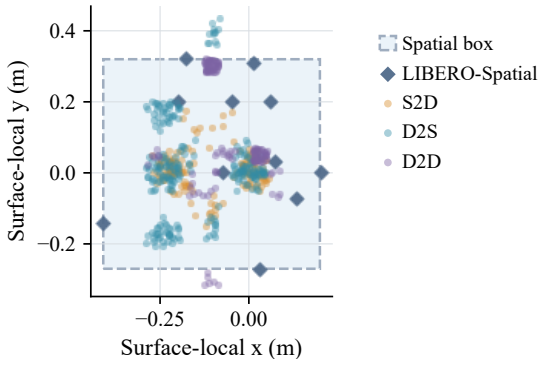


Figure 7: Declared placement-region centers. The dashed LIBERO-Spatial center range contains 510 of 528 Industry region declarations.

B.3 Language Instruction Schema

```
{
  "$schema": "http://json-schema.org/draft-07/schema#",
  "title": "IndustrialTaskInstructionSchema",
  "type": "object",
  "properties": {
    "action_verb": {
      "type": "string",
      "enum": [
        "pick and place",
```

```
        "unload",
        "route synchronous",
        "transfer"
      ]
    },
    "target_object": {
      "type": "string",
      "description": "Unique workpiece identifier."
    },
    "destination_fixture": {
      "type": "string",
      "description": "Destination fixture identifier."
    },
    "takt_constraint": {
      "type": "string",
      "pattern": "^within_[0-9]+_seconds$"
    }
  },
  "required": [
    "action_verb",
    "target_object",
    "destination_fixture",
    "takt_constraint"
  ]
}
```

B.4 Temporal Protocol and Operating Conditions

B.4.1 Terminology and Scope

Takt time is reserved for a demand-pacing requirement,

$$\tau_{\text{takt}} = \frac{T_{\text{available}}}{Q_{\text{demand}}}, \tag{9}$$

which relates available production time to required output (Linck & Cochran, 1999). Because the benchmark does not prescribe a customer demand scenario, its temporal settings are reported as arrival intervals, service-opportunity intervals, evaluation deadlines, and episode horizons. These quantities are not interchangeable: an episode horizon limits recording or rollout duration, whereas an arrival interval or service-opportunity interval specifies the workload presented to the controller.

Table 6: Temporal quantities used in LIBERO-INDUSTRY.

Symbol	Name	Operational meaning
τ_{takt}	Demand takt	Available production time per required product; used only when demand is specified.
τ_{src}	Source arrival interval	Time between consecutive source workpieces becoming available.
τ_{dst}	Destination interval	Time between consecutive destination service opportunities on a pulse-indexed line.
τ_d	Evaluation deadline	Latest admissible completion time for a scored workpiece.
T_{roll}	Episode horizon	Maximum simulated or recorded duration; not a production-rate claim.

B.4.2 Suite-Specific Physical Timing

For a continuous source conveyor with signed velocity V_s and consecutive workpiece pitch d_{src}, the source arrival interval is

$$\tau_{\text{src}} = \frac{d_{\text{src}}}{|V_s|}. \tag{10}$$

Equivalently, when an arrival interval is authored directly in the task definition, the corresponding pitch is $d_{\text{src}} = |V_s|\tau_{\text{src}}$. For a pulse-indexed destination conveyor, let T_{dwell} be the stationary service

phase and s_{slot} the distance advanced by one slot. Its service-opportunity interval is

$$\tau_{\text{dst}} = T_{\text{dwell}} + \frac{s_{\text{slot}}}{|V_d|}, \tag{11}$$

where V_d is the signed destination transfer velocity. The implementation derives s_{slot} from the fixture extent along the travel axis plus the declared inter-fixture clearance. The evaluated release records freeze the fixture asset, travel axis, clearance, V_d, and T_{dwell} for every task. S2D timing is therefore determined by the destination mechanism, not by an assumed robot movement time.

D2D tasks retain both source and destination constraints as the ordered pair

$$\boldsymbol{\tau}_i = (\tau_{\text{src},i}, \tau_{\text{dst},i}), \tag{12}$$

rather than collapsing two asynchronous lines into a single scalar interval. Table 7 summarizes the parameters that must be exported with the benchmark release.

Table 7: Timing provenance by task suite.

Suite	Source schedule	Destination schedule	Reported physical limits
D2S	V_s, d_{src}, τ_{src}, t_{off}	Static	τ_{src}
S2D	Static	V_d, s_{slot}, T_{dwell}	τ_{dst}
D2D	V_s, d_{src}, τ_{src}, t_{off}	V_d, s_{slot}, T_{dwell}	τ_{src}, τ_{dst}

The task-specific benchmark execution-time allowances are reported in Table 8.

B.4.3 Suite-Specific Episode Horizons

Equation equation 5 expands according to the transport events encoded by each suite. Each expansion reports both the continuous rollout duration and the corresponding integer number of control steps.

D2S. The horizon covers the source offset, all budgeted source-arrival intervals, one robot-cycle allowance, and the terminal buffer:

$$\begin{aligned} T_{\text{roll}}^{\text{D2S}} &= t_{\text{off}} + (N_s - 1)\tau_{\text{src}} + t_r + t_b, \\ K_{\text{roll}}^{\text{D2S}} &= \left\lceil f_c T_{\text{roll}}^{\text{D2S}} \right\rceil. \end{aligned} \tag{13}$$

S2D. The destination schedule must accommodate the slower of the robot cycle and the destination service interval for every target:

$$\begin{aligned} T_{\text{roll}}^{\text{S2D}} &= t_i + N_t \max(t_r, \tau_{\text{dst}}) + t_b, \\ K_{\text{roll}}^{\text{S2D}} &= \left\lceil f_c T_{\text{roll}}^{\text{S2D}} \right\rceil. \end{aligned} \tag{14}$$

D2D. The horizon combines the source-arrival schedule with one post-arrival allowance, one destination service interval, and the terminal buffer:

$$\begin{aligned} T_{\text{roll}}^{\text{D2D}} &= t_{\text{off}} + (N_s - 1)\tau_{\text{src}} + t_x + \tau_{\text{dst}} + t_b, \\ K_{\text{roll}}^{\text{D2D}} &= \left\lceil f_c T_{\text{roll}}^{\text{D2D}} \right\rceil. \end{aligned} \tag{15}$$

Here, N_s is the number of budgeted source positions; the generator uses the four-position source capacity for D2S and D2D Levels 2–3. N_t is the number of S2D targets. The BDDL value t_{off} is the time before the first source arrival, not a spatial offset. During initialization it is converted to an upstream displacement from the conveyor centre as $d_{\text{off}} = |V_s| t_{\text{off}}$. The fixed values are $t_{\text{off}} = 1.5\,\text{s}$, $t_i = 1.0\,\text{s}$, $t_r = 3.0\,\text{s}$, $t_x = 1.5\,\text{s}$, $t_b = 1.5\,\text{s}$, and $f_c = 20\,\text{Hz}$. The ceiling operation ensures that the discrete episode includes the complete continuous-time horizon.

Table 8: Suite-specific nominal benchmark takt and destination service-time settings.

Suite–Level	Takt @ V3/M3	Takt @ V5/M5	Takt @ V8/M8	Service Time
D2D–LV1	5.0 s	6.0 s	6.0 s	1.5 s
D2D–LV2	5.0 s	6.0 s	6.0 s	2.0 s
D2D–LV3	5.0 s	6.0 s	6.0 s	2.5 s
D2S–LV1	4.0 s	4.0 s	4.0 s	–
D2S–LV2	5.0 s	6.0 s	6.0 s	–
D2S–LV3	5.0 s	6.0 s	6.0 s	–
S2D–LV1	–	–	–	1.5 s
S2D–LV2	–	–	–	2.0 s
S2D–LV3	–	–	–	2.5 s

Note. V3/M3, V5/M5, and V8/M8 denote paired positive and negative source-conveyor speeds of ± 0.03, ± 0.05, and ± 0.08 m/s, respectively. Each table entry reports the nominal benchmark takt assigned at that speed condition.

B.4.4 Evaluation Freeze and Throughput Reporting

The comparison protocol freezes the task definition, timing condition, deadline grid, episode horizon, control frequency, random seeds, settling dwell, and success predicate before comparing policies. No timing value is retuned for an individual policy. Results report the complete success–deadline curve for each task, alongside the physical feasibility flags.

The theoretical opportunity rate associated with a scalar interval τ is

$$q_{\text{opportunity}} = \frac{3600}{\tau} \quad \text{opportunities/hour.} \tag{16}$$

Consequently, intervals of 4, 5, and 6 s correspond to 900, 720, and 600 theoretical opportunities/hour, respectively. They are not good-part throughput. Observed good-part throughput is instead

$$q_{\text{good}} = 3600\,\frac{N_{\text{successful}}}{T_{\text{observed}}} \quad \text{products/hour,} \tag{17}$$

and is reported as a benchmark-conditioned measurement rather than a universal HMLV production rate.

B.5 Conveyor Mechanics and Success Predicates

B.5.1 Simulation Mechanics and Transport Physics

The simulated transport system maintains physical continuity by updating the state vectors of ungrasped workpieces at every simulation step. Let $\mathbf{R}_B \in \mathbb{SO}(3)$ represent the rotation matrix of the conveyor mechanism in the world coordinate frame, $\hat{\mathbf{a}}_{\text{local}}$ the localized linear travel axis, and v_{belt} the scalar speed parameter. The instantaneous linear velocity update vector $\dot{\mathbf{q}}_{O_{xy}}(t)$ applied to the horizontal plane of workpiece O is formulated as:

$$\dot{\mathbf{q}}_{O_{xy}}(t) = \begin{cases} \mathbf{R}_B \cdot \hat{\mathbf{a}}_{\text{local}} \cdot v_{\text{belt}} & \text{if } \Phi_{\text{belt}}(O) \wedge \neg \Psi_{\text{grasp}}(O) \\ \dot{\mathbf{q}}_{O_{xy}}(t) & \text{otherwise} \end{cases} \tag{18}$$

where the Boolean predicate $\Phi_{\text{belt}}(O)$ indicates verified topological contact with the active conveyor mesh, and $\Psi_{\text{grasp}}(O)$ evaluates whether the robot's parallel jaw actuators have established an active, non-slip force closure on the workpiece geometry. Upon satisfying $\Phi_{\text{belt}}(O)$, an instant kinematic anchoring constraint synchronizes the object's base linear velocity directly to the transport vector, preventing non-linear drifting behavior outside explicit physics-driven slippage events.

B.5.2 Kinematic Overrides for Conveyor and Assembly Physics

To bypass simulation chattering and enforce high-rigidity industrial constraints, we apply explicit position and velocity overrides to dynamic objects at the MuJoCo runtime level. In discrete-time

physics engines, relying solely on surface friction to transport complex-mesh workpieces at high velocities causes unphysical micro-slippage and geometric interpenetration.

To address these limitations, Algorithm 1 dictates the initial conveyor transport kinematics. Specifically, the initial phase utilizes an instant kinematic anchoring mechanism to synchronize the workpiece with the belt upon establishing continuous geometric contact.

Algorithm 1 Conveyor Transport Kinematics (Starting Phase)

Input: sim.data, target_vel, obj_name, belt_geom_ids
is_contact $\leftarrow$ _in_contact_with(sim, obj_geom_ids, belt_geom_ids)
if not is_contact **then**
    return ▷ Allow standard gravity and free rigid-body dynamics
end if
qpos_adr, qvel_adr $\leftarrow$ GetJointAddresses(sim, obj_body_id)
if obj_name $\notin$ _anchored_quats **then**
    _anchored_quats[obj_name] $\leftarrow [1.0, 0.0, 0.0, 0.0]$
    _anchored_pos[obj_name] $\leftarrow$ sim.data.qpos[qpos_adr : qpos_adr + 2]
end if
▷ Enforce orientation lock to prevent mesh interpenetration jitter
sim.data.qpos[qpos_adr + 3 : qpos_adr + 7] $\leftarrow$ _anchored_quats[obj_name]
▷ Override velocities based on belt target speed
for $i \in \{0, 1\}$ **do**
    if |target_vel[i]| $<$ 1e-5 **then**
        sim.data.qpos[qpos_adr + i] $\leftarrow$ _anchored_pos[obj_name][i]
    end if
end for

Following the initial transport phase, Algorithm 2 governs the terminal insertion constraints into the moving conveyor slots. Subsequently, this terminal phase employs a multi-stage velocity arrest mechanism (Levels 1 to 3) to lock the workpiece inside the slot geometry upon satisfying strict dot-product orientation thresholds.

Algorithm 2 Multi-Stage Slot Arrest Kinematics (Ending Target)

Input: sim.data, vel_vec, delta_pos, lock_level
obj_qpos_adr, obj_qvel_adr $\leftarrow$ GetJointAddresses(sim, obj_body_id)
if AlignmentThresholdMet() **then**
    if lock_level $==$ 1 **then**
        ▷ Level 1: Planar Lock Only (X, Y locked to conveyor; Z/Rotation free)
        sim.data.qvel[obj_qvel_adr : obj_qvel_adr + 2] $\leftarrow$ vel_vec[0 : 2]
        sim.data.qpos[obj_qpos_adr : obj_qpos_adr + 2] += delta_pos[0 : 2]
    else
        ▷ Level 2 & 3: Full 3D Spatial Lock
        sim.data.qvel[obj_qvel_adr : obj_qvel_adr + 3] $\leftarrow$ vel_vec
        sim.data.qpos[obj_qpos_adr : obj_qpos_adr + 3] += delta_pos
        sim.data.qvel[obj_qvel_adr + 2] $\leftarrow$ 0.0 ▷ Arrest Z-axis gravity
    end if
end if
▷ Update moving groove target position continually
sim.data.qpos[qpos_adr : qpos_adr + 3] $\leftarrow$ exact_world_pos
sim.data.qpos[qpos_adr + 3 : qpos_adr + 7] $\leftarrow$ world_groove_quat_wxyz

B.5.3 Predicate Scaling and Constraint Matrix

To accommodate the progressive complexity of industrial workflows, the state evaluation predicates are stratified across three difficulty levels. As environmental variance increases, the task definitions

impose stricter kinematic constraints and require higher degrees of freedom (DoF) for successful execution.

Table 9: Kinematic Constraint Matrix by Predicate. A checkmark (✓) indicates the mechanism is strictly verified for task success, while a cross (×) indicates it is relaxed or not applicable.

Predicate	Gripper Released	Topological Contact	Vector Axis Alignment	Velocity Sync
`ExactOn`	✓	✓	×	×
`DynamicOn`	✓	✓	×	✓
`AlignedOn`	✓	✓	✓	×
`Routed`	✓	✓	✓	✓

The mapping of these target definition predicates across task paradigms is summarized in Table 10.

Table 10: Target Definition Predicates and Kinematic Constraints by Difficulty Level.

Difficulty Level	Receptacle Type	Static Target (D2S)	Dynamic Target (S2D, D2D)	Control Constraint
L1	Tray	`ExactOn`	`DynamicOn`	3-DoF
L2	Slot / V-Groove	`AlignedOn`	`Routed`	5-DoF
L3	Inlay	`AlignedOn`	`Routed`	6-DoF

B.6 Environmental Generalization Settings

B.6.1 Standards and Reporting Basis

The perturbation families are organized using selected industrial measurement and operating terminology. ISO 9283 provides the pose-accuracy and pose-repeatability framework used to describe bounded object, fixture, and camera-pose deviations (International Organization for Standardization, 1998). ISO 8995-1 provides the workplace-lighting context for controlled illumination changes (International Organization for Standardization, 2002), while GB/T 10595 provides the operating context for the commanded belt-conveyor conditions (Standardization Administration of China, 2017). These standards provide reporting context but neither prescribe nor validate the simulator-domain bounds below, which are fixed by the benchmark generator.

B.6.2 Perturbation Specifications

Level 1 is the nominal LIBERO-Industry condition. Levels 2 and 3 introduce moderate and broad generalization conditions, respectively.

Workspace Geometry. For object initialization, Level 1 scales each declared planar placement interval by 0.5 and fixes yaw. Levels 2 and 3 scale the interval by 0.2 and sample yaw from $\mathcal{U}(0, 2\pi)$. D2D fixture translations are bounded by ± 2 cm in Levels 1 and 2; Level 2 additionally applies a shared yaw perturbation of $\pm 5^\circ$. Level 3 applies cardinal transpositions, bounds translation to ± 2 cm in x and to ± 2 or ± 5 cm in y according to the transposition, and applies yaw perturbations of $\pm 5^\circ$ or $\pm 10^\circ$.

D2S Level 1 uses common fixture translations of up to ± 5 cm. Level 2 retains geometry-bounded translations of up to ± 5 cm and applies belt and groove yaw perturbations of $\pm 10^\circ$. Level 3 combines cardinal transpositions with common translations of up to ± 2 cm in x and ± 4 cm in y, 5 mm local inlay jitter, and yaw perturbations of $\pm 10^\circ$. S2D uses common translations of up to ± 5 cm in all tiers, adds independent source and destination yaw perturbations of $\pm 10^\circ$ at Level 2, and combines cardinal transpositions with $\pm 20^\circ$ yaw perturbations at Level 3.

Visual Observations. Table 11 reports the camera, illumination, and workbench parameters shared across D2D, D2S, and S2D tasks.

The illumination entries are normalized MuJoCo renderer parameters rather than photometric measurements in lux. They therefore provide reproducible simulator-domain conditions without asserting photometric conformity.

Table 11: Visual-observation perturbations across the three generalization tiers. Angular bounds are in degrees; translation bounds are Euclidean norms.

Configuration	Level 1: nominal	Level 2: moderate	Level 3: broad
Agent-view orbit	Azimuth 0; elevation 38.922340; distance 1.130651 m	Azimuth $\{45, 315\} \pm 5$; elevation 38.922340 ± 5; distance 1.130651 ± 0.1 m	Azimuth sampled from $[0, 135) \cup (230, 360)$; elevation 38.922340 ± 10; distance 1.130651 ± 0.2 m
Agent-view rotation	Fixed	Rotation norm ≤ 1	Rotation norm ≤ 2
Wrist camera	Not used	Translation norm ≤ 2.29 mm; rotation norm ≤ 1	Translation norm ≤ 4.57 mm; rotation norm ≤ 2
Illumination	Neutral profile; scale 2.5; no cast shadows	Neutral, warm, or cool profile; scale 2.5 ± 0.5; cast shadows	Neutral, warm, or cool profile; scale 2.5 ± 1.0; cast shadows
Workbench surface	Green matte	Green, blue, or black; matte or glossy	Same six controlled styles as Level 2

Transport Dynamics. The named conveyor-speed conditions are anchored at 0.03, 0.05, and 0.08 m/s. D2D samples magnitudes within the non-overlapping intervals $[0.0300, 0.0399]$, $[0.0400, 0.0649]$, and $[0.0650, 0.0800]$ m/s. Level 1 uses equal signed source and destination speeds, Level 2 uses unequal same-direction speeds, and Level 3 uses opposite directions. D2D service times are 1.5, 2.0, and 2.5 s for Levels 1–3.

For D2S, Level 1 applies $\pm 5\%$ relative speed variation around the selected condition, Level 2 applies ± 0.005 m/s absolute variation, and Level 3 uses the selected condition without additional speed noise. S2D uses the same Level 1 and Level 2 rules; Level 3 samples speed magnitude uniformly from $[0.03, 0.08]$ m/s while retaining the task direction. S2D service times are 1.5, 2.0, and 2.5 s for Levels 1–3.

B.6.3 Reproducibility and Configuration

All scene perturbations are deterministic for the task seed. The generator serializes the camera, lighting, surface, reset, and rollout parameters for each task in `libero/libero/benchmark/industry_env_args.json`. Suite-specific geometry, composition, and transport rules are defined in `notebooks/industry/autogen/configs.py`, `perturbations.py`, `task_composition.py`, and the D2D, D2S, and S2D generation scripts.

C Evaluation Protocol and Extended Results

C.1 Evaluated Models and Checkpoints

We evaluate five publicly released policies fine-tuned on the original LIBERO benchmark, without further fine-tuning on LIBERO-Industry. This is a cross-benchmark transfer evaluation, not an evaluation of unfine-tuned base models. Table 12 identifies the checkpoint sources. The exact weight revisions and model-specific inference settings should be recorded in the released evaluation configurations.

Table 12: Original-LIBERO-adapted checkpoints used for cross-benchmark evaluation.

Policy	Checkpoint identifier
$\pi_{0.5}$	`gs://openpi-assets/checkpoints/pi05_libero`
OpenVLA-OFT	`moojink/openvla-7b-oft-finetuned-libero-spatial-object-goal-10`
UniVLA	`qwbu/univla-7b-224-sft-libero`
X-VLA	`2toINF/X-VLA-Libero`
Cosmos Policy	`nvidia/Cosmos-Policy-LIBERO-Predict2-2B`

C.2 Evaluation-Set Breakdown

Table 13 reports the suite-level results for the same five checkpoints as Table 3.

Table 13: **Evaluation-set results by suite.** Scores in %; 120 rollouts per policy per suite.

	S2D			D2S			D2D		
Policy	SR	$\mathrm{PCR_T}$	$\mathrm{PCR_S}$	SR	$\mathrm{PCR_T}$	$\mathrm{PCR_S}$	SR	$\mathrm{PCR_T}$	$\mathrm{PCR_S}$
$\pi_{0.5}$	0.0	1.1	11.9	0.0	0.0	11.1	0.0	0.0	14.3
OpenVLA-OFT	0.0	0.0	9.1	0.0	0.0	11.0	0.0	0.0	13.4
UniVLA	0.0	0.0	10.8	0.0	0.0	11.1	0.0	0.0	14.7
X-VLA	0.0	0.0	10.8	0.0	0.0	11.1	0.0	0.0	13.8
Cosmos Policy	0.0	0.0	10.8	0.0	0.0	10.0	0.0	0.0	14.6

$\mathrm{PCR_S}$ includes initially valid destination workpieces (Section 3.5).

C.3 Cumulative Diagnostic Stress Progression

C.3.1 Stress-Progression Setup

Unless otherwise noted, the diagnostic progression uses the same evaluation setup as Section 4.1. Table 14 records the progression-specific task coverage, rollout counts, and language conditions.

Table 14: Task and language composition of the cumulative diagnostic stress progression.

Stage	**Registered tasks**	**Trials**	**Language condition**
D0	ABLD000–009	$10 \times 20 = 200$	Ten original spatial-relation prompts, such as "between the plate and the ramekin."
D1	ABLD100–109	$10 \times 20 = 200$	Generic prompt: "pick up the akita black bowl and place it on the plate."
D2	ABLD200–211	$12 \times 20 = 240$	Three singular D2D/D2S/S2D templates without a conveyor-source reference.
D3	ABLD300–311	$12 \times 20 = 240$	Singular family-specific templates aligned with the introduced conveyor geometry.
D4	ABLD400–411	$12 \times 20 = 240$	D2D and S2D introduce "moving tray"; D2S wording remains unchanged.
D5	ABLD500–511	$12 \times 20 = 240$	The D4 templates use plural object nouns.
D6	ABLD600–611	$12 \times 20 = 240$	The plural D5 instruction templates are retained.

C.3.2 Trajectory-Level Failure Classification

Figure 8 maps each rollout to the first unmet manipulation phase, from pre-reach policy freeze and reach failure through post-release goal failure. Reporting the tests and thresholds makes the labels auditable; the final evaluation must also freeze a precedence rule for episodes that satisfy more than one failure condition.

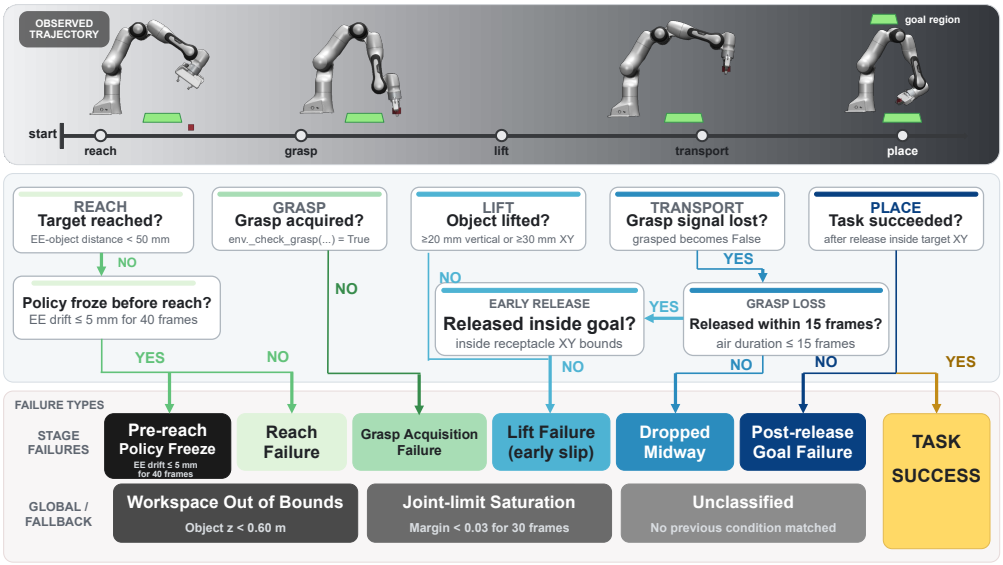


Figure 8: **Trajectory-level failure classification.** Each episode is evaluated along the reach–grasp–lift–transport–place sequence. The decision rules expose the operational thresholds used for stage-specific labels, while workspace violations, joint-limit saturation, and unmatched cases remain global or fallback outcomes.

The resulting labels are used in a cumulative diagnostic stress progression, not a one-factor-at-a-time causal ablation. From D0 to D6, the sequence adds an industrial background, removes a visual anchor, replaces the target with industrial trays, introduces a conveyor and temporal dynamics, and increases the sequence from one to three objects. Adjacent stages may change more than one factor, so their differences localize breakdowns without isolating causes. The historical values are not yet tied to frozen denominators, seeds, or confidence intervals. D6 has no successful trajectory, so its policy-step statistic is undefined.

C.3.3 Representative Failure Trajectories

To complement the aggregate failure proportions in Figure 3, Figure 9 shows representative frame sequences from four classified rollouts. Each row contains one trajectory from the frozen Pi0.5 tracker run and reports the corresponding stage-level frequency. The examples illustrate the operational meaning of the tracker outcomes; they are not an exhaustive validation of all classified episodes.

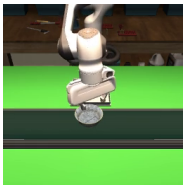
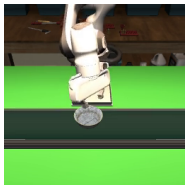
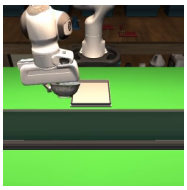
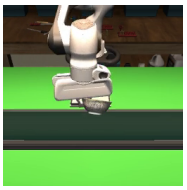
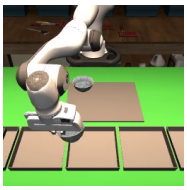
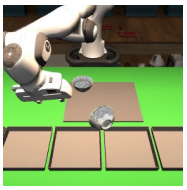
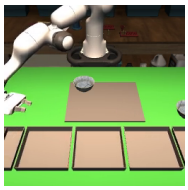
Failure phase	Representative trajectory frames			Tracker outcome
	t_1	t_2	t_3	
Spatial / reach D5; ABLD503; episode 14	 frame 0	 frame 189	 frame 399	**Spatial approach failure** 132/240 episodes (55.0%) at D5.
Grasp / lift D3; ABLD306; episode 10	 frame 42	 frame 147	 frame 399	**Kinematic lift failure** 9/240 episodes (3.8%) at D3.
Dynamic transport D4; ABLD406; episode 11	 frame 84	 frame 168	 frame 231	**Dynamic slip** Dropped midway; 35/240 episodes (14.6%) at D4.
Placement D5; ABLD509; episode 15	 frame 160	 frame 200	 frame 399	**Post-release goal failure** 28/240 episodes (11.7%) at D5.

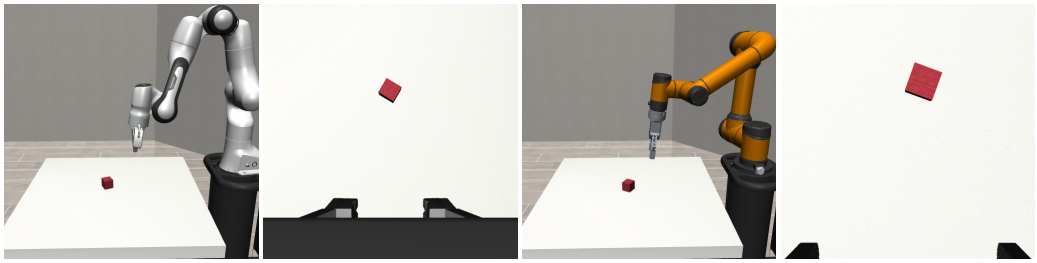
Figure 9: **Representative failure trajectories.** Each row shows three temporally ordered frames from one failed rollout, followed by the trajectory-level classifier output and its frequency within the corresponding diagnostic stage. The examples link the aggregate categories in Figure 3 to observable trajectory behavior and are illustrative rather than exhaustive.

D Cross-Embodiment Conversion and Policy Adaptation

Embodiment and task-space tracking. We instantiate an AuboI5 arm with a CTAG2F90D gripper, using the gripper's `grip_site` as the tool center point (TCP) provided by the factory provider. Panda and Aubo environments share the task definition and named non-robot scene state, while the Aubo arm and controller are initialized natively. After settling, corresponding task-body poses must agree within 10^{-6} m and 10^{-3} degrees. Because the arms have seven and six joints, respectively, conversion tracks Cartesian references rather than copying joint coordinates. In the shared simulator world frame, the Aubo action is

$$\mathbf{u}_t^A = \begin{bmatrix} \mathrm{clip}_{[-1,1]}\left(\begin{bmatrix} k(\mathbf{p}_{t+1}^P - \mathbf{p}_t^A)/s_p \\ k\,\mathrm{Log}\big(R_{t+1}^P (R_t^A)^\top\big)/s_R \end{bmatrix} \right) \\ g_t^P \end{bmatrix}, \tag{19}$$

where $k = 3$, $s_p = 0.05\,\text{m}$, $s_R = 0.5\,\text{rad}$, and Log returns a three-dimensional rotation vector. The target controller follows the next Panda TCP reference online while retaining the source gripper schedule. This aligns task-space references, not joint trajectories or pixelwise camera observations.



(a) Panda setup (b) Panda wrist (c) Aubo–CTAG setup (d) Aubo–CTAG wrist

Figure 10: Representative Lift-scene setup and wrist observations. The comparison checks robot–gripper composition and view coverage; it does not assume pixelwise camera equivalence.

Policy interface and visual convention. The policy receives external and wrist RGB images and a seven-dimensional state comprising TCP position, axis–angle orientation, and one CTAG driver coordinate. Its seven-dimensional action comprises six relative operational-space controller commands and one gripper command; no additional delta-action transform is applied. Images are stored at 256×256 and resized to 224×224 by OpenPI. For the aligned simulation releases, export and inference both use the same 180-degree rotation of raw simulator images. Since the recorded HDF5 images already contain a vertical flip, export applies a horizontal flip to obtain this convention.

D.1 Trajectory Conversion and Recovery

Motivation and scope. Motivated by MimicGen's demonstrated transfer of the same Panda demonstrations to Sawyer, IIWA, and UR5e (Mandlekar et al., 2023), we bridge the Panda-to-Aubo embodiment gap with a MimicGen-based recovery procedure that transforms object-centric end-effector segments into Aubo reference frames and retains only trajectories that pass three out of three replays. Our local implementation follows MimicGen's subtask decomposition and reference-frame transformation principles, but uses our own LIBERO and Aubo execution, validation, and data-materialization code rather than the released software.

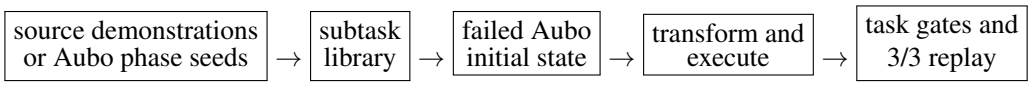


Figure 11: Recovery pipeline. A candidate is added to the training dataset only after generation and replay validation.

Recovery procedure. Gripper events and task predicates split successful source rollouts into object-centric segments. LIBERO-Object and LIBERO-Spatial use separate pick and place segments from successful Panda demonstrations, whereas LIBERO-Goal and LIBERO-10 use object, articulated-fixture, and push phases selected from successful Aubo phase seeds to preserve long-horizon task invariants. Given a source end-effector pose ${}^{W}T^{s}_{e,t}$ and the source and query reference poses ${}^{W}T^{s}_{r}$ and ${}^{W}T^{q}_{r}$, respectively, each segment is transformed as

$$ {}^{W}T^{q}_{e,t} = {}^{W}T^{q}_{r} \left({}^{W}T^{s}_{r}\right)^{-1} {}^{W}T^{s}_{e,t}, \tag{20} $$

which preserves the end-effector motion relative to the live object or fixture. Each attempt restores the failed conversion's exact Aubo initial state, bridges to the transformed segment by Cartesian interpolation, retains the demonstrated gripper schedule, and executes the poses with the Aubo operational-space controller. A candidate is retained only if its task predicates hold, its arrays

Table 15: Final Panda-to-Aubo conversion and recovery yield. Raw retained demonstrations passed all three exact-action replays. Recovery includes both direct conversion failures and replay-rejected raw demonstrations.

Suite	Raw 3/3 retained	Replay-rejected	Direct raw failures	Recovered 3/3	Final retained
LIBERO-Object	425	4	71	75	500
LIBERO-Spatial	147	38	315	353	500
LIBERO-Goal	216	41	243	284	500
LIBERO-10	184	37	279	316	500
Total	972	120	908	1,028	2,000

are finite and actions bounded, its joint-margin and controller-clipping checks pass, terminal success persists, and its stored actions succeed in three replays from the same initial state and scene. Accepted recoveries retain their demonstration identity and provenance and are merged only with replay-verified raw conversions, as summarized in Table 15.

E Physical System, Real-World Dataset, and Sim–Real Evaluation

E.1 Physical Workcell and Sensor Setup

The physical dynamic-to-static (D2S) workcell uses an AUBO-i5H arm with a CTAG2F90D gripper to transfer objects from a moving conveyor into stationary target fixtures. A Pico headset and controllers support teleoperated demonstration collection. An external HIKON camera and a wrist camera provide the two recorded RGB views (Figure 12); the policy's state and action interface is specified in Appendix D. The target fixture changes from a tray in Level 1 to object-matched grooves in Level 2 and form-fitting inlays in Level 3. The robot and two-view policy interface are shared across levels while placement constraints change at the destination.

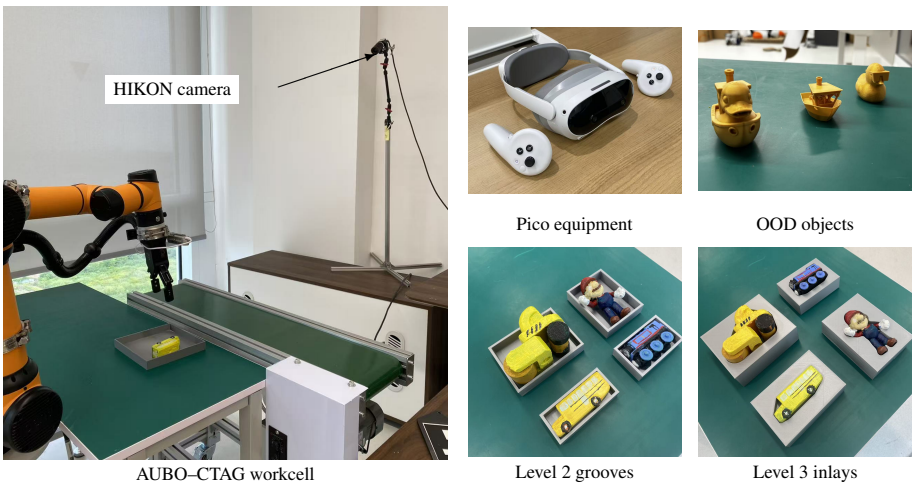


Figure 12: **Physical D2S workcell and fixtures.** Left: AUBO-i5H–CTAG arm, conveyor, target tray, and marked HIKON camera. Right, reading order: Pico teleoperation equipment, example OOD objects, Level 2 grooves, and Level 3 inlays.

E.2 Real-World Collection, Splits, and Quality Control

The physical study tests D2S transfer in the same robot and camera workcell across three levels. Level 1 uses tray targets; Level 2 introduces an initial-region or target-region confounder and uses matching grooves; Level 3 combines initial- and target-region confounders with form-fitting inlays. The levels probe pickup under scene variation and increasingly constrained target placement within

Table 16: Predeclared real-world training and held-out slot permutations. Letters denote School Bus (A), Mini Roller (B), Mario (C), and Thomas (D).

Combination	Training permutations	Held-out permutations
ABC	ACB, BAC, CAB, CBA	ABC, BCA
ABD	ABD, BAD, BDA, DAB	ADB, DBA
ACD	ACD, ADC, CDA, DCA	CAD, DAC
BCD	BDC, CBD, DBC, DCB	BCD, CDB

the same conveyor-to-stationary task structure. Because these factors change together, level differences do not isolate their individual effects. The physical collection design uses four three-object combinations. For each combination, four slot permutations are assigned to training and two disjoint permutations are reserved for held-out evaluation, as summarized in Table 16. A permutation specifies the left-to-right object order in the target slots. The planned training design contains

$$\begin{aligned} &4 \text{ combinations} \times 4 \text{ training permutations} \\ &\quad \times 2 \text{ directions} \times 3 \text{ disturbance levels} = 96 \text{ task configurations.} \end{aligned} \tag{21}$$

Conveyor speed is balanced across these configurations rather than fully crossed with every factor. A task configuration is distinct from a recorded trajectory; the accepted demonstration counts are given in Appendix E.3.

The held-out permutations are reserved from the physical training split.

E.3 Physical Demonstration Collection and Recorded Views

Figure 13 shows the Pico-assisted collection procedure. Figure 14 shows paired camera observations stored in an accepted HDF5 demonstration.

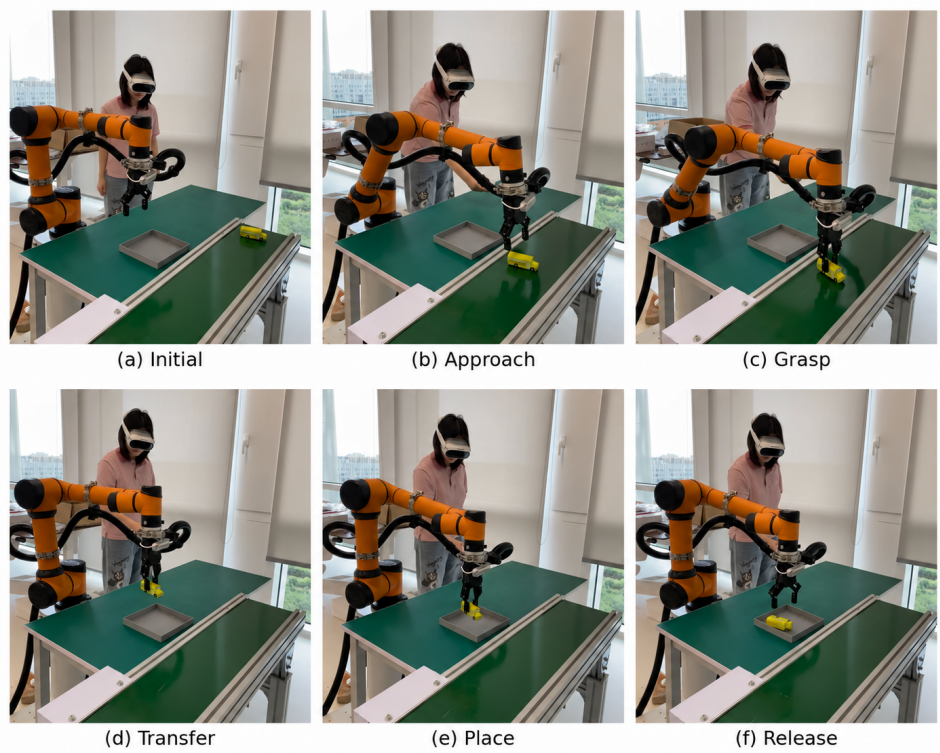


Figure 13: **Pico-assisted physical demonstration collection.** Six photographs illustrate the collection procedure: initial scene, approach, grasp, transfer, placement, and release for a Level 1 tray transfer.

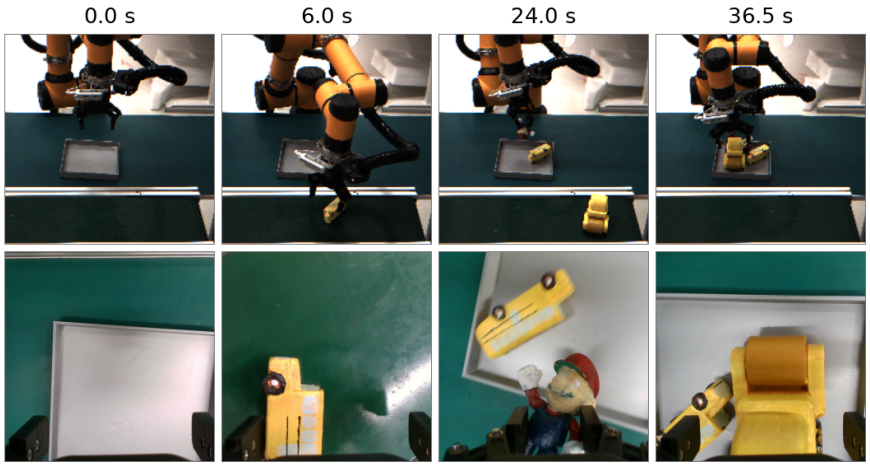


Figure 14: **Recorded physical demonstration `id_001`.** Stored language: "Retrieve the items from the conveyor and set them onto the plate." Each column pairs the table-camera view (top) with the wrist-camera view (bottom) at the same observation index of the aligned 10-Hz HDF5 episode. Times are relative to its first observation.

Accepted dataset. The accepted physical dataset contains 95 demonstrations and 41,257 frames. The accepted counts for Levels 1, 2, and 3 are 32, 32, and 31, respectively. Episode `id_092` was excluded because its nearest wrist-camera frame failed the predeclared 25 ms synchronization threshold. Both co-training conditions use this same accepted release.

E.4 SIM–REAL CO-TRAINING CONFIGURATION

Training conditions. Both runs load the same local copy of the released $\pi_{0.5}$ base weights (`pi05_base/params`) (Physical Intelligence, 2025b) and use the OpenPI training implementation (Physical Intelligence, 2025a). S1 combines Aubo-native original-LIBERO recovered-v2 simulation demonstrations with physical demonstrations. S2 replaces the simulation component with LIBERO-Industry demonstrations. Both conditions use the same 95 physical demonstrations. Table 17 summarizes the training sources. At physical evaluation, S1 and S2 face the same D2S workcell, object arrangements, target fixtures, and scoring rules. The comparison therefore tests the effect of changing the simulation training source under shared physical supervision; it does not attribute any difference to an individual confounder or fixture.

Training source	S1	S2
Simulation dataset	Original-LIBERO recovered-v2	LIBERO-Industry
Simulation episodes	1,724	1,412
Simulation frames	313,102	620,660
Physical episodes	95	95
Physical frames	41,257	41,257

Table 17: Training sources for the two co-training conditions. The physical dataset is shared, while the simulation source differs.

Sampling and optimization. Sampling operates on LeRobot frames using a 9:1 simulation-to-real ratio. This ratio specifies the sampling distribution rather than the proportion of unique trajectories. Both runs are configured for 30,000 optimizer updates with a global batch size of 252. The shared model and optimization settings are listed in Table 18. This comparison matches the physical-data source, sampling weights, and configured optimization budget, but not the sizes of the unique simulation datasets.

Setting	Shared configuration
Model	$\pi_{0.5}$
Base checkpoint	`pi05_base`
State input	Continuous
Action horizon	10
Simulation-to-real sampling	90:10, frame-based
Global batch size	252
Configured optimizer updates	30,000
Optimizer	AdamW
Gradient-norm clipping	1.0
Learning-rate schedule	Cosine schedule with warmup
Warmup steps	10,000
Peak learning rate	5×10^{-5}
Decay learning rate	5×10^{-5}
Scheduler decay steps	1,000,000
EMA decay	0.999
Training seed	42
Training hardware	Six NVIDIA A100 GPUs
FSDP devices	2
Data-loader workers	2
Checkpoint save interval	1,000 steps

Table 18: Shared configuration specified by the S1 and S2 launch records. The scheduler's decay-step parameter is distinct from the number of optimizer updates.

Checkpoint provenance. Training has completed for both S1 and S2. Each run was configured for 30,000 optimizer updates, with the final update indexed as `29999`.