TOWER: BENCHMARKING BIMANUAL MANIPULATION ACROSS SIMULATION AND REALITY

Anonymous authors
Paper under double-blind review

ABSTRACT

Complex robotic manipulation requires interpreting visual scenes and anticipating how actions change object contacts and structural support. Evaluation requires diverse, long-horizon tasks in which earlier physical interactions shape subsequent execution. To this end, we introduce **TOWER**, a novel Jenga-based benchmark for sustained contact and structural change, comprising six configurable task families involving bimanual coordination and tool use. Jenga provides repeatable structures in which local manipulations alter the support and stability needed for subsequent actions. A Real2Sim pipeline builds corresponding simulated environments for scalable evaluation. We evaluate four policy architectures across twelve configurations using separately trained real and simulated policies. Progress metrics reveal substantial partial completion hidden by binary task success, while taller configurations generally exhibit lower progress. Real–sim evaluation shows strong progress correlation (Pearson $r = 0.880$) and a normalized MMRV of 0.1217. TOWER provides a unified testbed for studying how policies respond to interaction-induced scene changes and supports systematic policy evaluation across reality and simulation. Code is available at `https://anonymous.4open.science/r/artifact-3d0e01332a1edd93`.

1 INTRODUCTION

Complex robotic manipulation requires interpreting visual scenes and accounting for how actions change object contacts and structural support. In scenes where objects support or constrain one another, a local manipulation can alter both the visible arrangement and the conditions for subsequent execution. A successful extraction can remove support, while a placement can change the stability of neighboring objects. Reliable policies must therefore adapt as interaction proceeds. Evaluating this capability calls for diverse, repeatable tasks in which earlier physical interactions shape the conditions for completing subsequent operations as object arrangements and support change.

We introduce TOWER, a novel Jenga-based benchmark for sustained contact and structural change in reality and simulation. Jenga provides a standardized physical foundation: regular block geometry supports repeatable configurations, while contact and support relationships couple local actions to later execution. We design six task families that vary manipulation goals, interaction types, and structural transformations, incorporating bimanual coordination and tool use where required. Transfer, extraction, insertion, and pushing produce different changes to the structures on which subsequent actions depend. Each family includes two- and three-layer configurations, yielding twelve evaluation configurations (Figure 2). This design enables comparisons across manipulation demands and within task families as tower structure and execution horizon change.

Systematic real-world evaluation requires repeated scene construction, resets, and policy rollouts, making broad comparisons costly (Li et al., 2025; Jain et al., 2025). To support scalable experimentation, we construct corresponding simulated environments through a Real2Sim pipeline. Scene reconstruction and camera alignment establish corresponding geometry and observations, while robot dynamics alignment and physics alignment ground simulated responses in physical measurements. The simulator enables repeated task-suite evaluation with shared definitions and criteria for task success, long-horizon progress, and non-target disturbance across the full suite of configurations.

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.

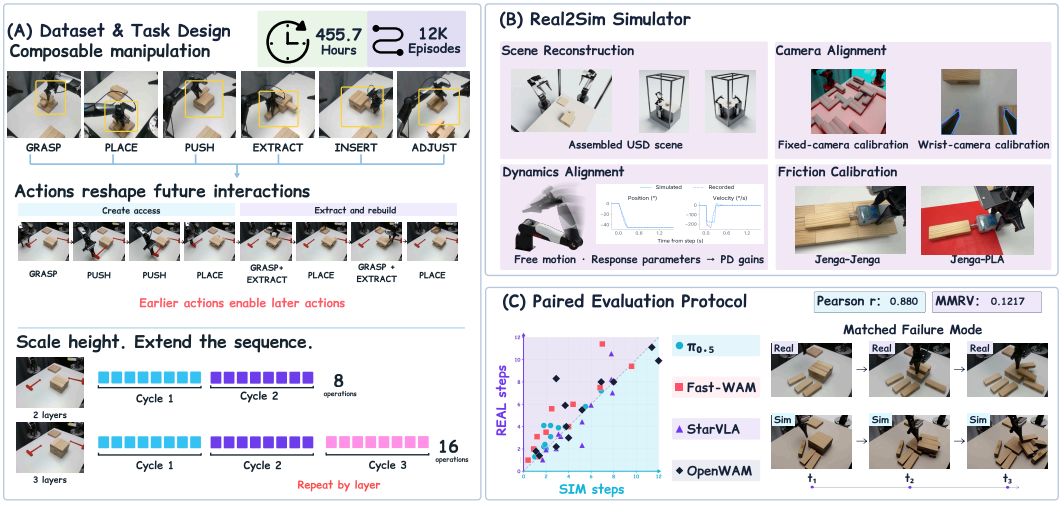


Figure 1: **Overview of TOWER.** (A) Dataset scale, composable manipulation primitives, and task sequences that reshape subsequent interactions and extend with tower height. (B) Scene reconstruction, camera alignment, dynamics alignment, and friction calibration. (C) Cross-domain progress correspondence, model comparisons, and matched examples of structural failure.

TOWER evaluates both progress towards the task goal and the physical consequences of execution. Binary task success cannot distinguish an early failure from a trial that completes most of its planned operations. Similarly, completing an operation does not indicate whether it disturbed neighboring blocks. We therefore report task completion, operation success, long-horizon progress, and non-target disturbance separately. Because some tasks intentionally reorganize the tower, the scoring distinguishes required structural changes from unintended effects on non-target objects. These measures describe task advancement and the accompanying changes to surrounding structures across the complete operation sequence, including changes that can affect the feasibility of later actions.

We evaluate four policy architectures across twelve configurations, yielding 48 paired real–sim settings. REAL-trained policies are evaluated in REAL, and SIM-trained policies in SIM, with matched architectures, task definitions, and scoring criteria. Long-Horizon Progress Rate (LHPR) ranges from 11.9% to 77.5% among configurations with zero Task Success Rate (TSR), revealing substantial variation hidden by binary outcomes. Progress decreases in five of six task families between the evaluated two- and three-layer configurations, as planned sequence length, interaction dependencies, and the underlying structural demands increase together within the task families.

Across domains, mean successful operation counts exhibit a Pearson correlation of 0.880, with normalized mean maximum rank violation (MMRV) of 0.1217. MMRV summarizes policy-ordering disagreements within matched task configurations. These findings characterize aggregate progress correspondence and policy-ordering differences between separately trained pipelines. TOWER provides a unified testbed for studying how policies respond to interaction-induced scene changes and supports systematic evaluation of manipulation policies across both reality and simulation.

- **A benchmark for sustained contact and structural change.** We design six Jenga-based task families spanning bimanual coordination, tool use, and dependent manipulation sequences, with twelve evaluation configurations, 12,000 training demonstrations, and shared scoring criteria for task completion, intermediate progress, and non-target disturbance.
- **A visually aligned, physics-grounded Real2Sim pipeline.** We combine scene reconstruction, camera alignment, measured robot-response fitting, and robot-held force-gauge measurements to construct visually and physically aligned simulated counterparts of the real task configurations.
- **A paired real-world and simulation study.** We examine progress beyond binary success, performance across tower configurations, and cross-domain correspondence in policy performance, rankings, and failure behavior across matched tasks and tower heights.

2 Related Work

Robot Manipulation Benchmarks ManiSkill2, VIMA, and Colosseum test diverse manipulation tasks and scene robustness (Gu et al., 2023; Jiang et al., 2023; Pumacay et al., 2024); RoboTwin 2.0 scales bimanual demonstrations and domain randomization (Chen et al., 2026). PerAct2, DuoBench, and BiCoord address two-arm coordination (Grotz et al., 2024; Jülg et al., 2026; Peng et al., 2026), while CALVIN, FurnitureBench, and FMB study sequential execution and assembly (Mees et al., 2022; Heo et al., 2023; Luo et al., 2025). Progress-aware measures complement terminal success (Peng et al., 2026; Wang et al., 2026a). Tower building and collapse-aware retrieval examine structural support (Noseworthy et al., 2021; Motoda et al., 2021; Pathak et al., 2026); robotic Jenga studies extraction and tower response (Wang et al., 2009; Fazeli et al., 2019; Puthuveetil et al., 2025), and BiCoord includes tower division and block transfer (Peng et al., 2026). TOWER combines bimanual and tool-assisted operations on configurable towers, assessing operation progress and disturbance as successive interactions change object arrangements and structural support.

Real-to-Sim Reconstruction and Physical Calibration RoboTwin generates assets and demonstrations, while PolaRiS combines reconstruction with simulation-data co-training (Mu et al., 2025; Jain et al., 2025). RobotArena∞ and SimFoundry reconstruct environments from demonstrations or video (Jangir et al., 2026; Ranawaka et al., 2026). SIMPLER and Tune to Learn fit control responses to recorded trajectories (Li et al., 2025; Bronars et al., 2026); other methods infer physical properties from visual, interaction, force, or tactile observations (Pfaff et al., 2025; Fan et al., 2025; Si et al., 2024). Contact-graph reconstruction and BrickSim represent support and connections (Xiang et al., 2026; Wen et al., 2026). Effective simulator parameters can differ from measured values (Acosta et al., 2022). TOWER combines scene and camera alignment, robot-response fitting, and force-gauge measurements to support paired policy evaluation in real and simulated environments.

Real–Sim Validation for Policy Evaluation SIMPLER evaluates correlation, ranking violations, and robustness (Li et al., 2025); REALM and SimFoundry examine task progression and policy-score correspondence (Sedlacek et al., 2026; Ranawaka et al., 2026). Training-domain choices affect these comparisons (Zhang et al., 2026a; Jain et al., 2025). Progress-aware scoring, matched-action replay, and motion comparisons characterize execution behavior (Jiang et al., 2026; Zhang et al., 2026a;b), while SureSim uses paired real trials to correct simulation bias and quantify uncertainty (Badithela et al., 2026). TOWER compares task completion, successful operations, disturbance, policy rankings, and failure behavior across matched task configurations.

3 TOWER Benchmark

3.1 Task Suite

Table 1: **Atomic actions (Figure 2).**

Action	Definition
Grasp	Securely hold a block or tool.
Place	Release an object stably at its target pose.
Push	Maintain contact to reach the required displacement or pose.
Extract	Pull a held target clear of its layer or slot.
Insert	Move into a slot to satisfy pose and depth tolerances.
Adjust	Correct a supported target's pose by contact.

TOWER comprises six task families with two- and three-layer variants, yielding 12 configurations. Figure 2 groups Tasks A–F by difficulty; Table 1 defines their action vocabulary. Appendix A.1 specifies scored G/P operations, while Appendix A.2 provides task definitions, scenario summaries (Table 3), and real-robot sequences.

Difficulty reflects horizon, tower height, contact constraints, and structural changes. Easy tasks relocate blocks; medium tasks require extraction or insertion; hard tasks combine ordered contact, tool use, and structural reconstruction.

3.2 Real-to-Sim Construction

TOWER constructs its simulator from the physical platform through a Real2Sim pipeline combining scene reconstruction, camera alignment, robot dynamics alignment, and physics alignment. Figure 3 summarizes these stages; implementation and calibration details are provided in Appendices C–E.

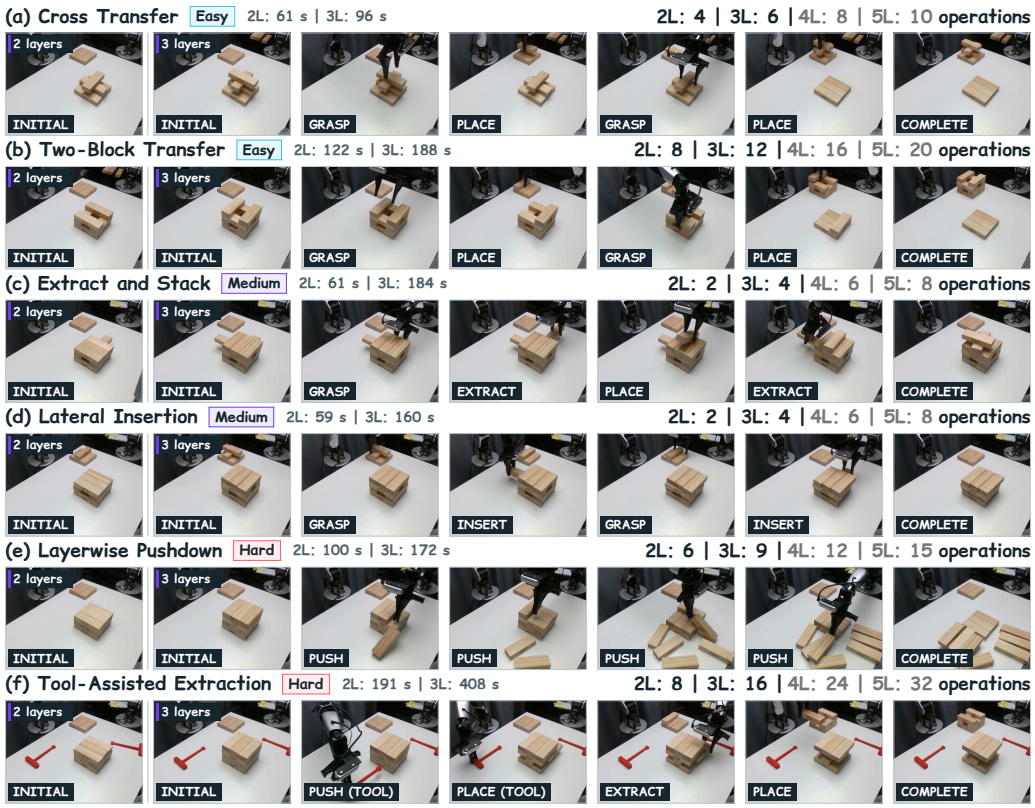


Figure 2: **TOWER task suite (Tasks A–F).** Each row shows the two- and three-layer initial configurations followed by representative real-world execution frames and a completed arrangement. Tasks A–B, C–D, and E–F form the easy, medium, and hard task categories, respectively.

Scene Reconstruction. Station CAD, photographs, and measured block properties define the simulated scene. An agent-assisted CAD-to-SimReady workflow converts the station into simulation assets. Photographs guide robot placement and surface appearance, while measured dimensions and mass define the physical properties of individual blocks in the assembled simulation environment.

Camera Alignment. Camera extrinsics are refined with fixed scene geometry and intrinsics. For the right-front static camera, a table-plane estimate and pose search initialize point-to-plane ICP:

$$(R^*, t^*) = \underset{R\in \mathrm{SO}(3),\, t\in\mathbb{R}^3}{\arg\min} \sum_i \left[n_i^\top (Rp_i + t - s_i)\right]^2, \tag{1}$$

where p_i is an observed depth point and s_i, n_i are its corresponding scene point and normal. Block silhouettes disambiguate symmetric matches before refinement of the camera pose:

$$T^* = \underset{T\in \mathrm{SE}(3)}{\arg\min} \left[\lambda_s \mathcal{L}_{\mathrm{sil}}(T) + \lambda_c \mathcal{L}_{\mathrm{contour}}(T) + \lambda_d \mathcal{L}_{\mathrm{depth}}(T)\right]. \tag{2}$$

The losses compare observed and rendered silhouettes, contours, and depth. The second static view verifies the resulting alignment against the geometry of the recorded real-world scene.

Wrist camera-to-gripper calibration. A rigid camera-to-gripper transform determines each wrist camera's pose as the arm moves. Matching silhouettes and contours across recorded robot poses calibrates the transform; Appendix E.2 provides the procedure and visual comparisons.

Robot Dynamics Alignment. Recorded step and sinusoidal motions calibrate each arm's response. We minimize the excited joint's position and velocity errors at the recorded timestamps:

$$\mathcal{L}_r(\theta) = \mathrm{MSE}(q_{\mathrm{sim}} - q_{\mathrm{real}}) + 0.05\ \mathrm{MSE}(\dot{q}_{\mathrm{sim}} - \dot{q}_{\mathrm{real}}). \tag{3}$$

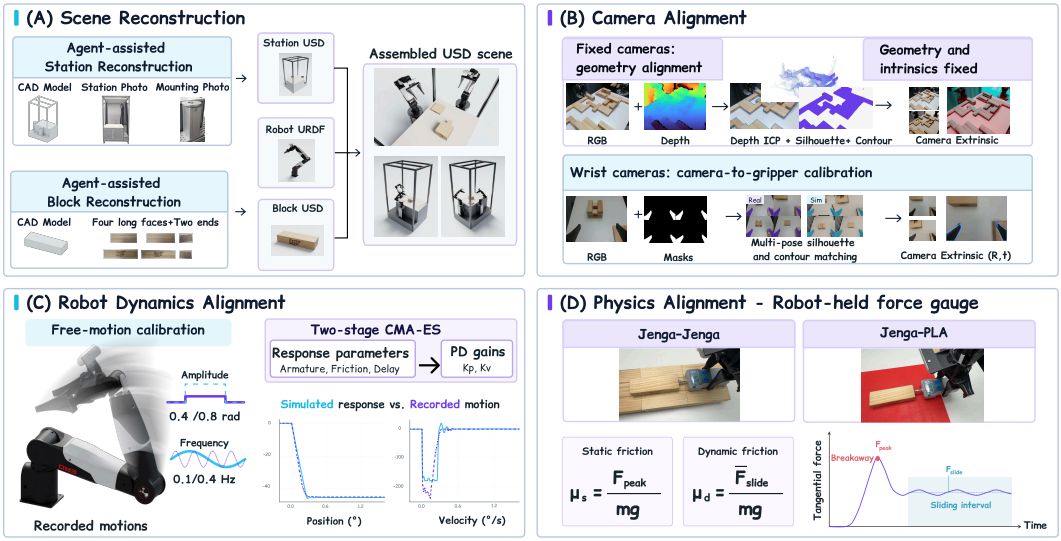


Figure 3: **Real2Sim construction.** (A) Station and block assets are reconstructed from CAD and photographs and assembled with the robot model. (B) Fixed and wrist cameras are aligned through geometry and silhouette matching. (C) Two-stage CMA-ES fits response parameters and PD gains; curves compare simulated and recorded joint motion. (D) Force-gauge measurements characterize Jenga–Jenga and Jenga–PLA friction; the schematic force profile shows breakaway and sliding.

Two-stage CMA-ES fits joint armatures, viscous and Coulomb friction, and command delay before proportional and derivative gains. Held-out position RMSE decreases by 28.0% and 18.5% for the left and back arms, respectively; velocity RMSE decreases by 45.2% and 51.1%. The fitted response parameters and command delays are applied during SIM evaluation (Appendix E.4).

Physics Alignment. A robot-held force gauge measures static and sliding friction for Jenga–Jenga and Jenga–PLA contact. For normal load mg, breakaway and mean sliding forces determine the coefficients, and paired contacting materials use the average material-combination rule:

$$\mu_s = \frac{F_{\text{peak}}}{mg}, \qquad \mu_d = \frac{\overline{F}_{\text{slide}}}{mg}, \qquad \mu_{AB} = \frac{\mu_A + \mu_B}{2}. \tag{4}$$

where F_{peak} and $\overline{F}_{\text{slide}}$ denote breakaway and mean sliding forces. Block coefficients of $0.551/0.263$ (static/dynamic) are applied in SIM; Appendix E.3 details the material estimates. These stages address complementary components of the interaction: camera alignment establishes the observation geometry, response fitting matches robot motion, and friction calibration characterizes resistance at object contacts. Together, they define the simulated counterpart used for the shared task suite and evaluation protocol across all six tasks and both tower heights.

3.3 Evaluation Protocol and Metrics

TOWER evaluates both task achievement and the disturbances caused during execution. Five metrics capture task completion, operation success, long-horizon progress, and non-target disturbance. Scores are computed separately in REAL and SIM over N valid episodes per model–task–layer configuration. Two annotators assess each episode, and a third reviewer resolves disagreements. The simulator implements an automatic scorer checked against human-annotated SIM evaluation episodes. All reported results use human annotations; Appendix B details scoring and agreement.

Tasks comprise predefined G and P operations: grasping and placing for transfer tasks, with task-specific definitions for pushing and tool use (Appendix A). For episode i, N_i^G, N_i^P denote planned counts and S_i^G, S_i^P successful counts. Each planned operation earns credit at most once. Failed attempts and unstarted operations receive no credit in the reported operation counts.

Task completion. Task Success Rate (TSR) measures completion of all prescribed operations with a stable final arrangement satisfying the task criteria. With binary episode success s_i,

$$\mathrm{TSR} = \frac{1}{N}\sum_{i=1}^{N} s_i. \tag{5}$$

Supplementary operation success. The main analysis focuses on TSR, LHPR, and ODR; complete task-level metrics are provided in Appendix G.4. Grasp Success Rate (GSR) and Place Success Rate (PSR) divide successful G and P counts by their respective planned totals:

$$\mathrm{GSR} = \frac{\sum_{i=1}^{N} S_i^G}{\sum_{i=1}^{N} N_i^G}, \qquad \mathrm{PSR} = \frac{\sum_{i=1}^{N} S_i^P}{\sum_{i=1}^{N} N_i^P}. \tag{6}$$

Standard transfers require a secure grasp and stable placement after release; other operations follow their task-specific criteria. The denominators include all planned operations, even unstarted ones.

Long-horizon progress. Long-Horizon Progress Rate (LHPR) measures successful operations relative to the full planned sequence. With $C_i = S_i^G + S_i^P$,

$$\mathrm{LHPR} = \frac{\sum_{i=1}^{N} C_i}{\sum_{i=1}^{N} (N_i^G + N_i^P)}. \tag{7}$$

Full operation progress can still fail the final-state criterion, so LHPR is reported alongside TSR. LHPR credits successful planned operations without requiring an uninterrupted successful prefix, while retaining unstarted operations in the planned total used as its denominator.

Non-target disturbance. Operation Disturbance Rate (ODR) measures non-target block-disturbance occurrences relative to a conventionally counted step total:

$$E_i = C_i + (1 - s_i), \qquad \mathrm{ODR} = \frac{\sum_{i=1}^{N} D_i}{\sum_{i=1}^{N} E_i}, \tag{8}$$

where D_i counts disturbances, once per non-target block per operation, with repeated counting across operations. The denominator counts successful operations plus one for an unsuccessful episode; it does not count actual attempts or retries. Lower ODR is better; values can exceed one. Model comparisons use the same configuration set; missing results are not replaced with zero.

4 Experiments

4.1 Experimental Setup

Evaluated policies. We evaluate two VLAs, $\pi_{0.5}$ (Black et al., 2025) and StarVLA-α (Ye et al., 2026), and two WAMs, Fast-WAM (Yuan et al., 2026) and OpenWAM (Wang et al., 2026b), using four RGB views, a task instruction, and model-specific state inputs and action adapters. Each task–layer configuration has separate REAL- and SIM-trained checkpoints, evaluated in their respective domains. Comparisons match architectures, tasks, and scoring criteria (Appendix G.1).

Training settings. Each task–layer configuration provides 500 training demonstrations and two held-out validation demonstrations per domain, shared across policies. REAL demonstrations are collected through teleoperation, and SIM demonstrations through scripted task controllers.

Execution schedule. Both domains execute 16 predicted targets before acquiring observations and requesting the next prediction. Episodes have a 420 s limit; SIM lifts remain fixed for all policies. Appendix D.3 describes the action interface, Appendix F details demonstration collection, and Appendix G.2 specifies the evaluation cohort, trial counts, and metric aggregation rules. The balanced evaluation contains 960 episodes: ten trials for every policy, task, layer count, and domain. Each domain therefore contributes 480 episodes, giving all configurations equal representation in the task-level comparisons under shared definitions of task success and operation progress.

4.2 SIM–REAL CORRESPONDENCE

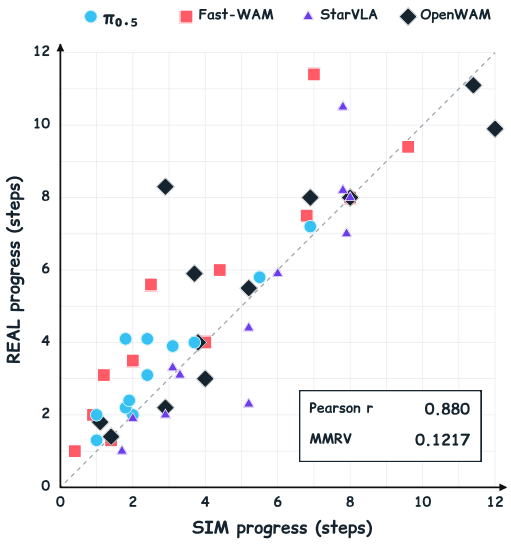


Figure 4: **SIM–REAL progress.** Points pair policy–task–layer means (10 episodes per domain). Colors/shapes identify policies; the dashed line marks equality. Appendix G.3 defines the metrics and aggregation rules.

Cross-domain progress. We compare progress, disturbance, and failure patterns across domains. The analysis covers four policies across Tasks A–F at two and three layers, yielding 48 paired configuration means. Each mean summarizes ten episodes for one policy–task–layer setting in one domain. Figure 4 shows a strong positive correlation between SIM and REAL progress (Pearson $r = 0.880$), with a mean absolute gap of 0.994 successful operations per episode. Configurations that achieve greater progress in SIM generally also advance further in REAL. These results support cross-domain correspondence in long-horizon performance across policies trained and evaluated separately in the two domains.

Policy ordering. Normalized MMRV is 0.1217. It compares policies within each task–layer configuration, using progress normalized by the planned operation count, and summarizes the real-performance margins associated with ordering disagreements. Lower values indicate better ordering agreement. Pearson describes the association across all 48 means, whereas MMRV describes policy ordering within each configuration.

The absolute gap measures the cross-domain difference in mean successful operation counts. Together, these measures characterize correspondence in progress and ranking; correlation itself does not imply equal performance. Appendix G.3 gives the computation and aggregation rules.

Model-level disturbance. Progress and disturbance characterize complementary aspects of execution. Table 2 pools 120 episodes per policy and domain across the same twelve task–layer configurations to compare non-target disturbance using the same counted-operation convention.

Table 2: **Operation disturbance rate (ODR).** Values pool 120 episodes per policy and domain across Tasks A–F at two and three layers. Lower is better; bold marks the lowest ODR per domain. Ranks indicate SIM→REAL ordering.

Policy	SIM↓	REAL↓	Rank
OpenWAM	0.386	**0.296**	2 → 1
StarVLA-α	**0.369**	0.428	1 → 2
Fast-WAM	0.502	0.493	3 → 3
$\pi_{0.5}$	0.851	0.556	4 → 4

SIM and REAL agree on five of six pairwise ODR orderings. OpenWAM and StarVLA-α exchange the top two positions; Fast-WAM and $\pi_{0.5}$ remain third and fourth. Fast-WAM also has similar ODR values, 0.502 and 0.493. Lower ODR means fewer disturbances per counted step (Section 3.3). Totals are pooled before division, rather than averaging configuration-level ratios. Rank agreement and numerical agreement capture different aspects of this comparison: $\pi_{0.5}$ retains fourth place although its ODR changes from 0.851 in SIM to 0.556 in REAL. Fast-WAM retains third place with much closer values. Reading scores and ranks together distinguishes relative disturbance ordering from agreement in magnitude.

Cross-domain failure patterns. Figure 5 presents six representative modes: insertion failure, push-induced instability, placement/release failure, extraction-induced disturbance, grasp failure, and tool-push instability. Both SIM and REAL exhibit these phenomena. Within the displayed categories, insertion failure is more frequent in SIM than REAL (8.13% versus 5.42%), whereas placement/release failure is more frequent in REAL (4.17% versus 2.29%). These shared modes provide behavioral evidence alongside the progress analysis; Appendix H gives classification details. The examples distinguish failure to establish the intended contact from loss of stability after contact. This distinction connects operation-level failure to the evolving tower state: grasping and insertion depend on access and alignment, while pushing, extraction, and release can change neighboring support and the contact geometry available to subsequent operations in the planned sequence.

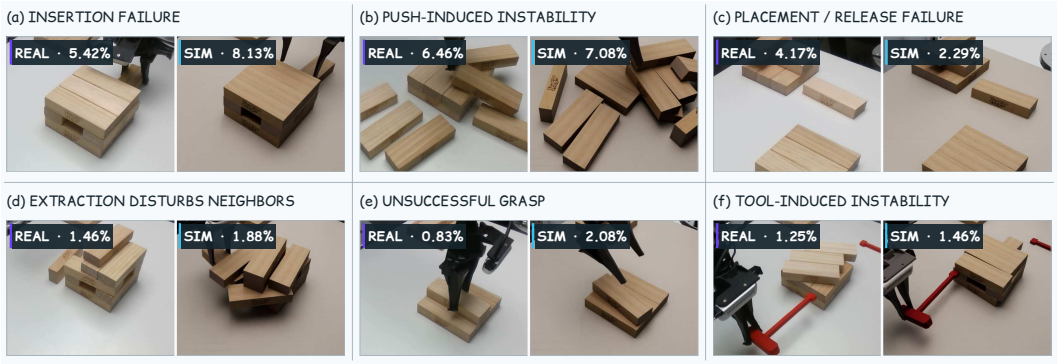


Figure 5: **Representative SIM–REAL failure modes.** Each panel pairs REAL (left) and SIM (right) examples from the same task. Percentages report provisional failure-mode prevalence among all episodes per domain; matched examples may be drawn from different evaluated policies.

Structural evolution and subsequent execution. Operations change both their targets and the conditions for subsequent actions. Pushing can displace neighbors, extraction changes contacts and support, and placement transfers support from gripper to tower. Thus, completing an operation does not preserve the conditions for later execution. Policies must use post-interaction observations to adapt as gaps, contacts, and support relationships evolve, linking the observed failure modes to sustained manipulation as interactions alter the surrounding structure throughout execution.

4.3 Beyond Binary Success: Long-Horizon Progress

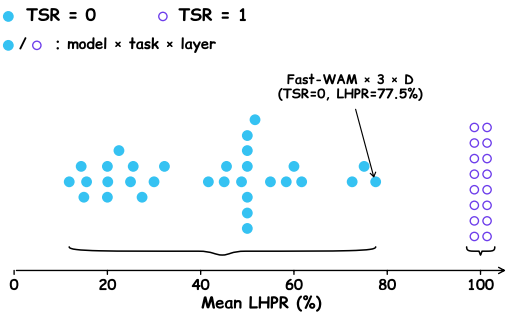


Figure 6: **Progress at matched TSR.** Each point averages 10 episodes. Cyan filled/purple open points denote TSR$=0/1$; both open columns correspond to 100% LHPR.

TOWER combines its manipulation primitives into dependent sequences. TSR treats early termination and failure near completion alike. LHPR instead measures the fraction of planned scored operations completed under each task's G/P scheme, distinguishing progress at the same success rate. Each policy–task–layer setting contains ten episodes per evaluation domain.

Progress despite failure. The 31 zero-TSR configurations span 11.9%–77.5% LHPR. Fast-WAM completes 77.5% of planned operations on REAL Task D–3L despite failing all ten episodes. All 16 TSR$=1$ settings reach 100% LHPR. These extremes distinguish complete execution from measurable partial operation progress.

Figure 6 shows how LHPR differentiates progress at matched TSR. For REAL D–3L, 77.5% LHPR corresponds to 3.1 of four planned operations per episode, although none of the ten trials completes the task. Appendix G.4, Tables 14–17, reports complete metrics and episode-level variability for all configurations, including separate REAL/SIM results and raw-count standard deviations.

Complementary measures. LHPR measures completed planned operations; ODR measures non-target disturbance. Similar progress can accompany different disturbance levels, while low disturbance need not imply substantial progress. Together with the cases, these metrics characterize task advancement and effects on surrounding structures, connecting SIM–REAL comparison to both goal-directed progress and the preservation of non-target objects and their supporting structures.

4.4 Scaling Task Difficulty with Tower Height

TOWER scales tasks through tower height and planned operation sequences. Figure 7 compares two- and three-layer configurations, averaging equally over four policies and both domains.

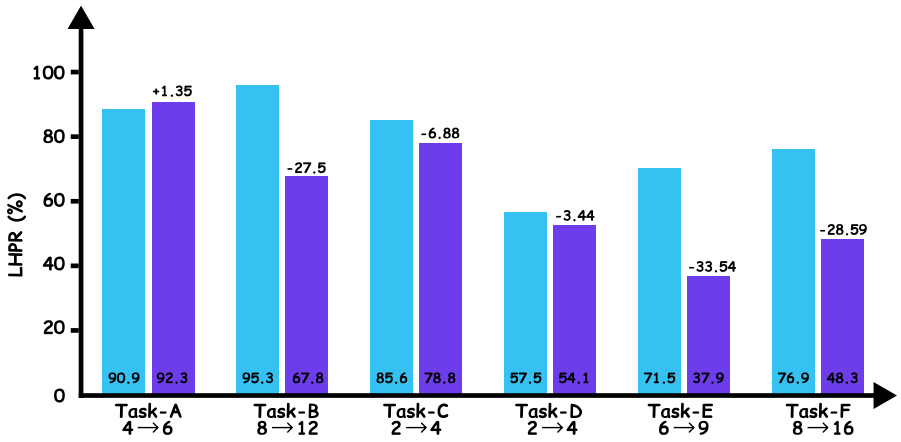


Figure 7: **LHPR across tower heights.** Cyan/purple bars denote two/three layers. Each averages eight policy–domain configurations with 10 episodes each. Upper labels show changes in percentage points; lower labels specify the planned operation counts for both tower heights.

Mean LHPR falls from 79.62% to 63.19%, with decreases in five of six tasks. B, E, and F show the largest drops (27.50, 33.54, and 28.59 percentage points) as their sequences grow from 8, 6, and 8 operations to 12, 9, and 16. Because LHPR is normalized by the planned operation count, these decreases indicate that policies complete a smaller fraction of the longer sequences, rather than simply reflecting an increase in the total number of operations required by the task.

The distinction between operation count and completion fraction is visible in Tasks B and F. For B, mean successful operations increase from 7.625 to 8.138 as the plan grows from 8 to 12 operations; for F, they increase from 6.150 to 7.725 as the plan doubles from 8 to 16. Policies thus perform more successful operations in these taller configurations while completing a smaller share of the task. Reporting LHPR alongside the planned sequence length makes this scaling effect explicit.

Task A remains high at 90.94% and 92.29% LHPR. C and D increase from two to four planned operations, with smaller drops of 6.88 and 3.44 percentage points. Longer sequences introduce more dependencies, and earlier deviations may affect subsequent execution. The contrast across tasks shows why height should be considered together with operation count and task structure. A larger tower need not reduce progress uniformly: transfer sequences, constrained insertion, and tool-assisted extraction impose different dependencies on earlier actions. The platform supports towers up to 18 layers, providing scope to extend these comparisons to longer operation sequences and varied contact and support configurations under the same LHPR scoring scheme.

5 Conclusion

We introduced TOWER, a benchmark for sustained bimanual manipulation in which actions reshape subsequent interactions. Six Jenga-based task families combine manipulation primitives into dependent sequences across two tower heights. A measurement-grounded Real2Sim pipeline and shared metrics for task completion, progress, and non-target disturbance support comparison across real and simulated environments as object configurations and support relationships evolve.

Evaluation of four policies across 48 paired settings shows strong correspondence in mean successful operation counts, with Pearson $r = 0.880$ and normalized MMRV of 0.1217. Model-level ODR rankings agree except for an exchange of the top two policies, while shared failure modes provide complementary behavioral evidence. LHPR further distinguishes substantial progress among configurations with zero task success and captures reduced completion fractions in five of six tasks at greater tower height. Together, these findings support evaluating final outcomes, intermediate progress, and structural effects through composable tasks of varying scale. Future work will study taller towers, broader material properties, and policy transfer between reality and simulation.

AI USE STATEMENT

Generative AI assisted writing, translation, literature searches, methodological discussions, result interpretation, code development, and figure drafting. Final figures were manually recreated. AI did not generate demonstrations, clean evaluation records, or assign labels. The authors reviewed AI-assisted content, verified references, code, and results, and take responsibility for the final paper.

REFERENCES

Brian Acosta, William Yang, and Michael Posa. Validating robotics simulators on real-world impacts. *IEEE Robotics and Automation Letters*, 2022. doi: 10.1109/LRA.2022.3174367. URL https://doi.org/10.1109/LRA.2022.3174367.

Apurva Badithela, David Snyder, Lihan Zha, Joseph Mikhail, Matthew O'Kelly, Anushri Dixit, and Anirudha Majumdar. Reliable and scalable robot policy evaluation with imperfect simulators. In *IEEE International Conference on Robotics and Automation*, 2026. URL https://ras.papercept.net/conferences/conferences/ICRA26/program/ICRA26_ContentListWeb_5.html.

Kevin Black, Noah Brown, James Darpinian, Karan Dhabalia, Danny Driess, Adnan Esmail, Michael Robert 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. In *Proceedings of the 9th Conference on Robot Learning*, volume 305 of *Proceedings of Machine Learning Research*, pp. 17–40. PMLR, 2025. URL https://proceedings.mlr.press/v305/black25a.html.

Antonia Bronars, Younghyo Park, and Pulkit Agrawal. Tune to learn: How controller gains shape robot policy learning. In *Robotics: Science and Systems*, 2026. doi: 10.15607/RSS.2026.XXII.139. URL https://www.roboticsproceedings.org/rss22/p139.html.

Tianxing Chen, Zanxin Chen, Baijun Chen, Zijian Cai, Yibin Liu, Zixuan Li, Qiwei Liang, Xianliang Lin, Yiheng Ge, Zhenyu Gu, Weiliang Deng, Yubin Guo, Tian Nian, Xuanbing Xie, Qiangyu Chen, Kailun Su, Tianling Xu, Guodong Liu, Mengkang Hu, Huan ang Gao, Kaixuan Wang, Zhixuan Liang, Yusen Qin, Xiaokang Yang, Ping Luo, and Yao Mu. RoboTwin 2.0: A Scalable Data Generator and Benchmark with Strong Domain Randomization for Robust Bimanual Robotic Manipulation. In *International Conference on Machine Learning*, 2026. URL https://icml.cc/virtual/2026/poster/62192.

Hongwei Fan, Hang Dai, Jiyao Zhang, Jinzhou Li, Qiyang Yan, Yujie Zhao, Mingju Gao, Jinghang Wu, Hao Tang, and Hao Dong. Twinaligner: Visual-dynamic alignment empowers physics-aware real2sim2real for robotic manipulation, 2025. URL https://arxiv.org/abs/2512.19390v1.

Nima Fazeli, Miquel Oller, Jiajun Wu, Zheng Wu, Joshua B. Tenenbaum, and Alberto Rodriguez. See, Feel, Act: Hierarchical Learning for Complex Manipulation Skills with Multisensory Fusion. Science Robotics 4(26), 2019, 2019. URL https://doi.org/10.1126/scirobotics.aav3123.

Markus Grotz, Mohit Shridhar, Tamim Asfour, and Dieter Fox. PerAct2: Benchmarking and Learning for Robotic Bimanual Manipulation Tasks. arXiv preprint, 2024. URL https://arxiv.org/abs/2407.00278v2.

Jiayuan Gu, Fanbo Xiang, Xuanlin Li, Zhan Ling, Xiqiang Liu, Tongzhou Mu, Yihe Tang, Stone Tao, Xinyue Wei, Yunchao Yao, Xiaodi Yuan, Pengwei Xie, Zhiao Huang, Rui Chen, and Hao Su. ManiSkill2: A Unified Benchmark for Generalizable Manipulation Skills. In *International Conference on Learning Representations*, 2023. URL https://openreview.net/pdf/a180ff20bf660bc9f8a2f007193ed71e83ba91f7.pdf.

Minho Heo, Youngwoon Lee, Doohyun Lee, and Joseph J. Lim. FurnitureBench: Reproducible Real-World Benchmark for Long-Horizon Complex Manipulation. In *Robotics: Science and Systems*, 2023. doi: 10.15607/RSS.2023.XIX.041. URL https://doi.org/10.15607/RSS.2023.XIX.041.

Arhan Jain, Mingtong Zhang, Kanav Arora, William Chen, Marcel Torne, Muhammad Zubair Irshad, Sergey Zakharov, Yue Wang, Sergey Levine, Chelsea Finn, Wei-Chiu Ma, Dhruv Shah, Abhishek Gupta, and Karl Pertsch. PolaRiS: Scalable real-to-sim evaluations for generalist robot policies. *arXiv preprint arXiv:2512.16881*, 2025. URL https://arxiv.org/abs/2512.16881.

Yash Jangir, Yidi Zhang, Pang-Chi Lo, Kashu Yamazaki, Chenyu Zhang, Kuan-Hsun Tu, Tsung-Wei Ke, Lei Ke, Yonatan Bisk, and Katerina Fragkiadaki. Robotarena∞: Scalable robot benchmarking via real-to-sim translation. In *International Conference on Learning Representations*, 2026. URL https://openreview.net/pdf?id=OutljIofvS.

Guangqi Jiang, Haoran Chang, Ri-Zhao Qiu, Yutong Liang, Mazeyu Ji, Jiyue Zhu, Zhao Dong, Xueyan Zou, and Xiaolong Wang. Gsworld: Closed-loop photo-realistic simulation suite for robotic manipulation. In *IEEE International Conference on Robotics and Automation*, 2026. URL https://3dgsworld.github.io/.

Yunfan Jiang, Agrim Gupta, Zichen Zhang, Guanzhi Wang, Yongqiang Dou, Yanjun Chen, Li Fei-Fei, Anima Anandkumar, Yuke Zhu, and Linxi Fan. VIMA: Robot Manipulation with Multimodal Prompts. In *Proceedings of the 40th International Conference on Machine Learning*, volume 202 of *Proceedings of Machine Learning Research*. PMLR, 2023. URL https://proceedings.mlr.press/v202/jiang23b.html.

Tobias Jülg, Seongjin Bien, Simon Hilber, Yannik Blei, Pierre Krack, Maximilian Li, Sven Parusel, Rudolf Lioutikov, Florian Walter, and Wolfram Burgard. DuoBench: A Reproducible Benchmark for Bimanual Manipulation in Simulation and the Real World. arXiv preprint, 2026. URL https://arxiv.org/abs/2606.11901v1.

Xuanlin Li, Kyle Hsu, Jiayuan Gu, Oier Mees, Karl Pertsch, Homer Rich Walke, Chuyuan Fu, Ishikaa Lunawat, Isabel Sieh, Sean Kirmani, Sergey Levine, Jiajun Wu, Chelsea Finn, Hao Su, Quan Vuong, and Ted Xiao. Evaluating real-world robot manipulation policies in simulation. In *Proceedings of The 8th Conference on Robot Learning*, volume 270 of *Proceedings of Machine Learning Research*, pp. 3705–3728. PMLR, 2025. URL https://proceedings.mlr.press/v270/li25c.html.

Jianlan Luo, Charles Xu, Fangchen Liu, Liam Tan, Zipeng Lin, Jeffrey Wu, Pieter Abbeel, and Sergey Levine. FMB: A Functional Manipulation Benchmark for Generalizable Robotic Learning. *The International Journal of Robotics Research*, 44(4), 2025. doi: 10.1177/02783649241276017. URL https://doi.org/10.1177/02783649241276017.

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. doi: 10.1109/LRA.2022.3180108. URL https://doi.org/10.1109/LRA.2022.3180108.

Tomohiro Motoda, Damien Petit, Weiwei Wan, and Kensuke Harada. Bimanual shelf picking planner based on collapse prediction. In *IEEE International Conference on Automation Science and Engineering*, pp. 510–515, 2021. doi: 10.1109/CASE49439.2021.9551507. URL https://doi.org/10.1109/CASE49439.2021.9551507.

Yao Mu, Tianxing Chen, Zanxin Chen, Shijia Peng, Zhiqian Lan, Zeyu Gao, Zhixuan Liang, Qiaojun Yu, Yude Zou, Mingkun Xu, Lunkai Lin, Zhiqiang Xie, Mingyu Ding, and Ping Luo. RoboTwin: Dual-Arm Robot Benchmark with Generative Digital Twins. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 27649–27660, 2025. URL https://openaccess.thecvf.com/content/CVPR2025/html/Mu_RoboTwin_Dual-Arm_Robot_Benchmark_with_Generative_Digital_Twins_CVPR_2025_paper.html.

Michael Noseworthy, Caris Moses, Isaiah Brand, Sebastian Castro, Leslie Kaelbling, Tomás Lozano-Pérez, and Nicholas Roy. Active learning of abstract plan feasibility. In *Robotics: Science and Systems XVII*, 2021. doi: 10.15607/RSS.2021.XVII.043. URL https://roboticsproceedings.org/rss17/p043.html.

Abhinav Pathak, Kalaichelvi Venkatesan, Tarek Taha, and Rajkumar Muthusamy. Collapse and collision aware grasping for cluttered shelf picking. *Frontiers in Robotics and AI*, 13:1697561, 2026. doi: 10.3389/frobt.2026.1697561. URL https://doi.org/10.3389/frobt.2026.1697561.

Xingyu Peng, Chen Gao, Liankai Jin, Annan Li, and Si Liu. BiCoord: A Bimanual Manipulation Benchmark towards Long-Horizon Spatial-Temporal Coordination. arXiv preprint, 2026. URL https://arxiv.org/abs/2604.05831v1.

Nicholas Pfaff, Evelyn Fu, Jeremy Binagia, Phillip Isola, and Russ Tedrake. Scalable Real2Sim: Physics-Aware Asset Generation Via Robotic Pick-and-Place Setups. In *IEEE/RSJ International Conference on Intelligent Robots and Systems*, 2025. URL https://scalable-real2sim.github.io/.

Wilbert Pumacay, Ishika Singh, Jiafei Duan, Ranjay Krishna, Jesse Thomason, and Dieter Fox. THE COLOSSEUM: A Benchmark for Evaluating Generalization for Robotic Manipulation. In *Robotics: Science and Systems*, 2024. doi: 10.15607/RSS.2024.XX.133. URL https://www.roboticsproceedings.org/rss20/p133.html.

Kavya Puthuveetil, Xinyi Zhang, Kazuto Yokoyama, and Tetsuya Narita. Strategic Jenga Play via Graph Based Dynamics Modeling. ICRA Workshop on Learning Meets Model-Based Methods for Contact-Rich Manipulation, 2025. URL https://contact-rich.github.io/assets/pdf/papers/24_Strategic_Jenga_Play_via_Graph.pdf.

Nadun Ranawaka, Josiah Wong, Wei-Lin Pai, Wei-Teng Chu, Tianyuan Dai, Masoud Moghani, Hang Yin, Yunfan Jiang, Wesley Durbano, Brandon Huynh, Yu Fang, Danfei Xu, Ruohan Zhang, Fei-Fei Li, Linxi Fan, Bowen Wen, Ajay Mandlekar, and Yuke Zhu. Simfoundry: Modular and automated scene generation for policy learning and evaluation, 2026. URL https://arxiv.org/abs/2606.28276.

Martin Sedlacek, Pavlo Yefanov, Georgy Ponimatkin, Jai Bardhan, Simon Pilc, Mederic Fourmy, Evangelos Kazakos, Cees G. M. Snoek, Josef Sivic, and Vladimir Petrik. Realm: A real-to-sim validated benchmark for generalization in robotic manipulation. *IEEE Robotics and Automation Letters*, 2026. URL https://martin-sedlacek.com/realm/. Accepted for publication.

Zilin Si, Gu Zhang, Qingwei Ben, Branden Romero, Zhou Xian, Chao Liu, and Chuang Gan. Difftactile: A physics-based differentiable tactile simulator for contact-rich robotic manipulation. In *International Conference on Learning Representations*, 2024. URL https://openreview.net/pdf?id=eJHnSg783t.

Jiuguang Wang, Philip Rogers, Lonnie Parker, Douglas Brooks, and Mike Stilman. Robot Jenga: Autonomous and Strategic Block Extraction. In *IEEE/RSJ International Conference on Intelligent Robots and Systems*, 2009. doi: 10.1109/IROS.2009.5354303. URL https://repository.gatech.edu/entities/publication/b242f6a1-9891-4bc4-b2f3-bd0cde18f163.

Yi Ru Wang, Carter Ung, Christopher Tan, Grant Tannert, Jiafei Duan, Josephine Li, Anh Le, Rishabh Oswal, Markus Grotz, Wilbert Pumacay, Yuquan Deng, Ranjay Krishna, Dieter Fox, and Siddhartha Srinivasa. RoboEval: Where Robotic Manipulation Meets Structured and Scalable Evaluation. In *IEEE International Conference on Robotics and Automation*, 2026a. URL https://ras.papercept.net/conferences/conferences/ICRA26/program/ICRA26_ContentListWeb_5.html.

Yuran Wang, Siqiao Huang, Mingleyang Li, Chenhao Zhang, Jiaqi Liang, Weiyang Jin, Yue Chen, Xuemin Chi, Donghao Zhou, Qize Yu, Yu-Kai Wang, Yuhan Rui, Shenzhe Yao, Zhen Yuan, Zhenhao Shen, Kefei Zhu, Zijie Zhu, Ning Gao, Xiaowei Chi, Guanqi He, Shanghang Zhang, Hao

Dong, Lin Shao, and Hang Zhao. OpenWAM: An open, modular exploration towards systematic world-action model pretraining. *arXiv preprint arXiv:2609.07398*, 2026b. URL https://arxiv.org/abs/2609.07398.

Haowei Wen, Ruixuan Liu, Weiyi Piao, Siyu Li, and Changliu Liu. BrickSim: A Physics-Based Simulator for Manipulating Interlocking Brick Assemblies, 2026. URL https://arxiv.org/abs/2603.16853v1.

Tianyi Xiang, Jiahang Cao, Sikai Guo, Guoyang Zhao, Andrew F. Luo, and Jun Ma. Real-to-Sim for Highly Cluttered Environments via Physics-Consistent Inter-Object Reasoning. *IEEE Robotics and Automation Letters*, 2026. URL https://ieeexplore.ieee.org/abstract/document/11543205.

Jinhui Ye, Ning Gao, Senqiao Yang, Jinliang Zheng, Zixuan Wang, Yuxin Chen, Pengguang Chen, Yilun Chen, Shu Liu, and Jiaya Jia. StarVLA-α: Reducing complexity in vision-language-action systems. In *European Conference on Computer Vision*, 2026. URL https://arxiv.org/abs/2604.11757. Accepted; author manuscript.

Tianyuan Yuan, Zibin Dong, Yicheng Liu, and Hang Zhao. Fast-WAM: Do world action models need test-time future imagination? *arXiv preprint arXiv:2603.16666*, 2026. URL https://arxiv.org/abs/2603.16666.

Kaifeng Zhang, Shuo Sha, Hanxiao Jiang, Matthew Loper, Hyunjong Song, Guangyan Cai, Zhuo Xu, Xiaochen Hu, Changxi Zheng, and Yunzhu Li. Real-to-sim robot policy evaluation with gaussian splatting simulation of soft-body interactions. In *IEEE International Conference on Robotics and Automation*, 2026a. URL https://real2sim-eval.github.io/.

Shujie Zhang, Jingkun Yi, Weipeng Zhong, Zirui Zhou, Yangkun Zhu, Hanqing Wang, Xudong Xu, Weinan Zhang, and Chunhua Shen. Robosnap: One-shot real-to-sim scene generation for generalizable robot learning and evaluation, 2026b. URL https://arxiv.org/abs/2607.06699v1.

APPENDIX CONTENTS

A Benchmark Specification **16**

A.1 Atomic Action Vocabulary 16

A.2 Task Definitions and Visual Sequences 16

A.3 Task Demonstration Collection and Annotation Protocol 19

A.4 Stepwise Evaluation Procedure 19

B Success, Failure, and Annotation Criteria **20**

B.1 Objective Episode Criteria 20

B.2 Human Annotation and Sampled Agreement 21

C Real-World Platform **22**

C.1 Robot and Control System 23

C.2 Camera System 23

C.3 Blocks, Foundations, and Initial-State Construction 23

D Simulation Environment **24**

D.1 Simulator and Scene Implementation 24

D.2 Physics Parameters 24

D.3 Robot Controller and Action Execution 25

D.4 Simulation Initialization and Randomization 26

E Real-to-Sim Alignment **26**

E.1 Scene Reconstruction 26

E.2 Camera Alignment 27

E.3 Physics Alignment 29

E.4 Robot-Response Calibration 30

F Dataset and Demonstration Collection **31**

F.1 Demonstration Collection 31

F.2 Unified Data Representation 31

F.3 Data Construction and Split 32

G Complete Evaluation Protocol **32**

G.1 Training–Evaluation Domain Matrix 32

G.2 Trial Counts and Metric Implementation 33

G.3 Statistical Analysis 33

G.4 Complete Metrics and Episode-Level Variability 34

H Failure Analysis **38**

H.1 Screening and Failure Definitions 38
H.2 Selection and Quantitative Results 39
H.3 Representative Cases 40

I Reproducibility and Evaluation Scope **40**
I.1 Reproducibility 40
I.2 Evaluation Scope 42

A BENCHMARK SPECIFICATION

A.1 ATOMIC ACTION VOCABULARY

We describe task execution using six semantic action primitives that visibly change an object or its relation to the tower. Intermediate motion phases—including approach, lift, free-space transport, pre-alignment, gripper opening, and retraction—are included within the corresponding semantic primitive and are not counted independently. Table 1 in Section 3.1 fixes the English names, meanings, and completion boundaries used throughout the task definitions, demonstration annotations, and evaluation protocol.

The primitives describe *what the robot does*; G- and P-group steps describe *what the evaluation counts*. A G step covers the task-specific acquisition or controlled-interaction unit, whereas a P step covers its task-specific terminal placement, insertion, tool return, or pose-adjustment unit. A counting step may therefore contain more than one primitive: for example, GRASP→EXTRACT forms one G step when extraction is the acquisition objective. The mapping is fixed before collection and is stated once in Section A.3.

Figure 8 shows three temporally ordered real-robot frames for every primitive: action onset, action in progress, and the resulting state. All images use the left-back camera, and the yellow box marks only the interaction region.

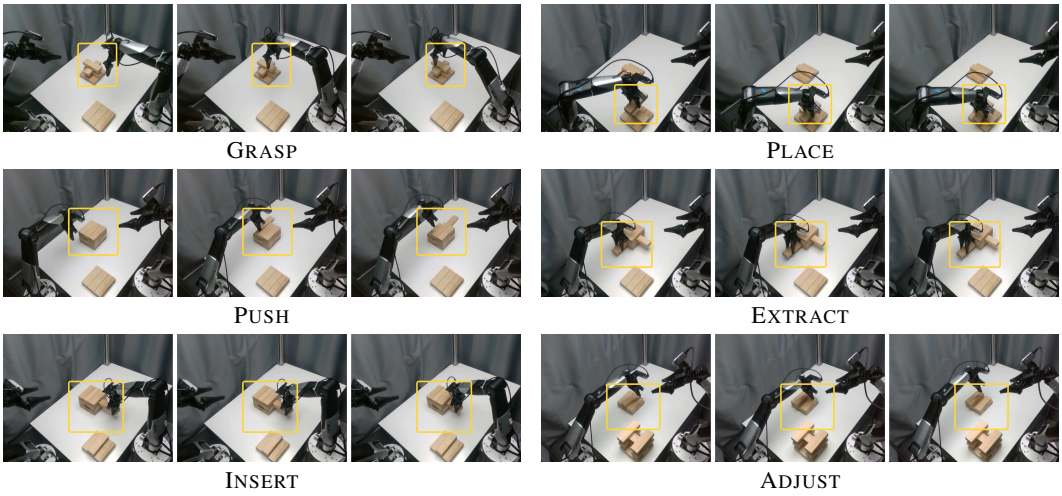


Figure 8: Three-frame examples of the six atomic action primitives.

A.2 TASK DEFINITIONS AND VISUAL SEQUENCES

Table 3 names the six task families and their difficulty groups. Figures 9–14 then show the initial state, one representative intermediate operation, and the final state of the complete task. Each task is paired with its own three-frame figure. All panels are real-robot observations from the left-back camera; the yellow box marks the representative operation region rather than an object mask.

Table 3: The six task families and their difficulty groups. Each family has two- and three-layer variants.

Difficulty	Task	Scenario
Easy	Task A: Cross Transfer	Cross-oriented single-block tower relocation.
	Task B: Two-Block Transfer	Paired-block relocation with alternating layers.
Medium	Task C: Extract and Stack	Extract center blocks and stack them on top.
	Task D: Lateral Insertion	Insert blocks into constrained center slots.
Hard	Task E: Layerwise Pushdown	Push designated blocks in a fixed layerwise order.
	Task F: Tool-Assisted Extraction	Tool-assisted extraction and ordered relocation.

Task A: Cross Transfer. Each source layer contains one target block, and adjacent layers alternate by approximately $90°$. The robot transfers blocks one by one and reconstructs the layer order and cross orientation on the destination base. Success requires every block to be transferred with the prescribed orientation and the reconstructed tower to remain stable.

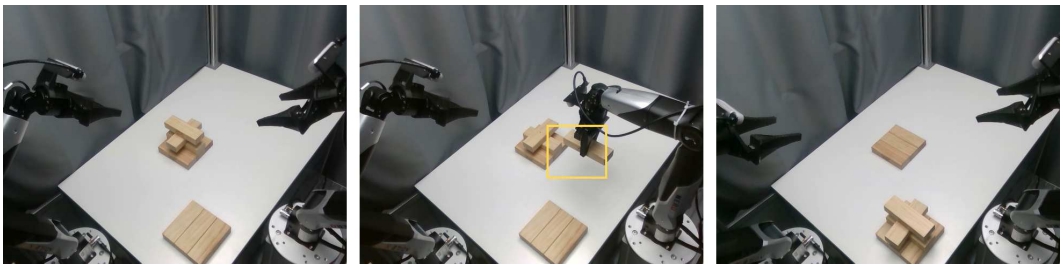


Figure 9: **Task A: Cross Transfer.** Initial state, representative transfer, and complete-task final state.

Task B: Two-Block Transfer. Each layer contains two side blocks, with adjacent layers alternating in orientation. The robot must preserve both the within-layer spacing and the layer-to-layer orientation while rebuilding the tower at a new location. Errors in an early placement change the support surface for every subsequent block.

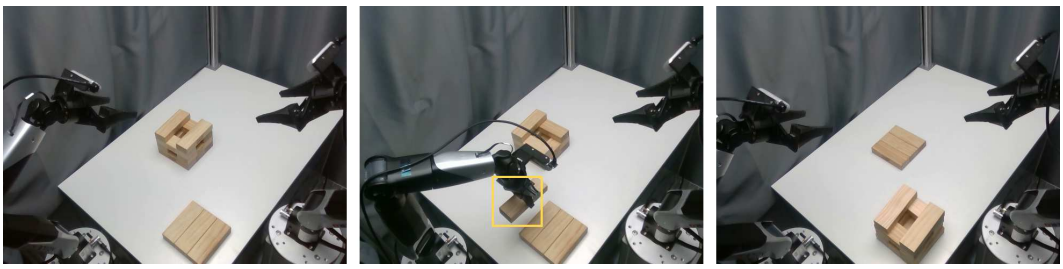


Figure 10: **Task B: Two-Block Transfer.** Initial state, representative transfer, and complete-task final state.

Task C: Extract and Stack. The target center block is initially protruded to expose a limited grasping surface. The robot grasps the exposed end, extracts the block horizontally without destabilizing neighboring supports, and places it at the prescribed tower-top pose. The operation combines constrained extraction with a stability-sensitive placement.

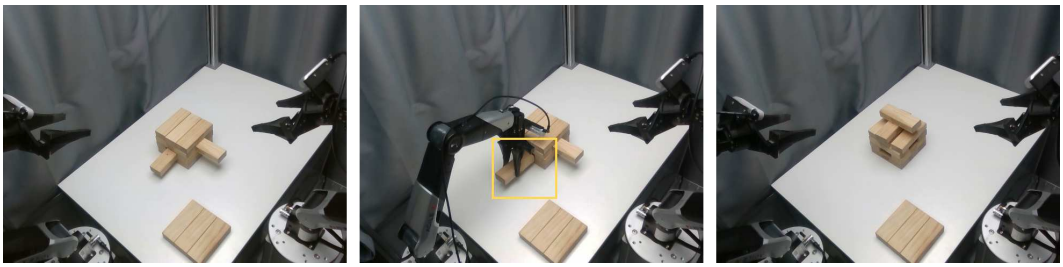


Figure 11: **Task C: Extract and Stack.** Initial state, representative extraction, and complete-task final state.

Task D: Lateral Insertion. The robot grasps an additional block, aligns it with the narrow gap between two side blocks, inserts it, and pushes it to the required depth. Small angular errors can jam the block or disturb the surrounding structure.

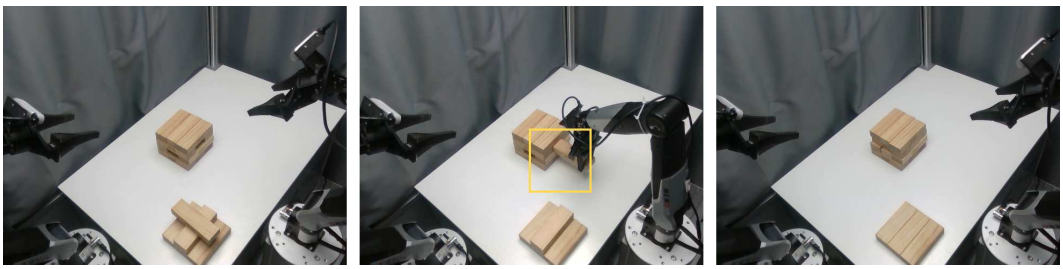


Figure 12: **Task D: Lateral Insertion.** Initial state, representative insertion, and complete-task final state.

Task E: Layerwise Pushdown. Starting at the top of a full tower, the robot pushes designated blocks according to the prescribed arm-specific order before proceeding to the next layer. Each contact changes the structure presented to later actions. Premature motion of an unprocessed block is recorded as non-target disturbance.

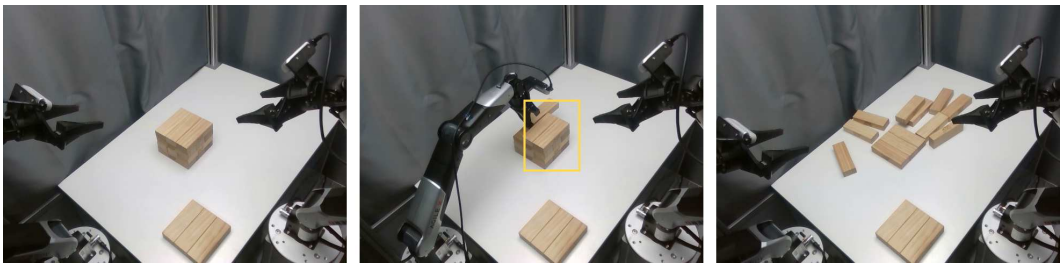


Figure 13: **Task E: Layerwise Pushdown.** Initial state, representative push, and complete-task final state.

Task F: Tool-Assisted Extraction. One arm grasps a dedicated red tool, pushes two designated side blocks into graspable configurations, and returns the tool. The other arm then extracts the exposed blocks and places them in the destination region. The task combines tool use, sequential bimanual roles, and execution under changing scene geometry.

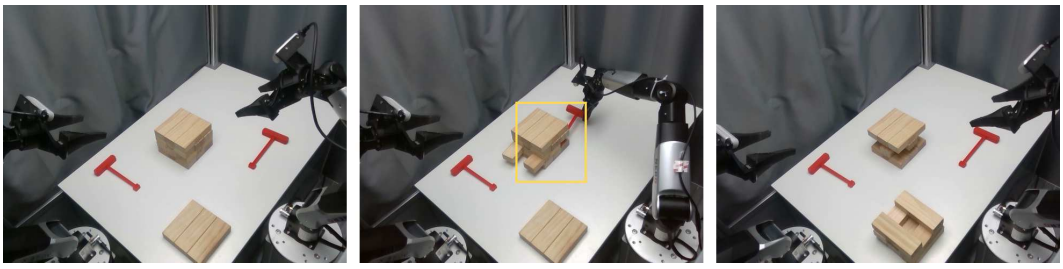


Figure 14: **Task F: Tool-Assisted Extraction.** Initial state, representative tool-assisted push, and complete-task final state.

A.3 Task Demonstration Collection and Annotation Protocol

Each demonstration begins from the task-specific canonical reset and is assigned a two- or three-layer configuration before recording starts. Synchronized observations and robot commands are recorded continuously from the initial state until the complete task reaches its final stability check or the episode is terminated. The operator follows the fixed primitive sequence for that configuration. Any change in order, omitted target, or recovery action is logged explicitly for quality review rather than being silently absorbed into a neighboring primitive.

Every semantic action is annotated using the vocabulary in Table 1, together with the manipulated object, its parent G/P step, start and completion boundaries, success flag, failure reason, and the identities of disturbed non-target blocks. Table 4 gives the canonical collection sequence and its G/P assignment. The two count columns cover all 12 task–layer configurations. We omit a separate total column because the LHPR denominator is always $N^G + N^P$.

Table 4: Canonical collection sequences and fixed G/P counts for the 12 task configurations. Entries x/y denote $N^G = x$ and $N^P = y$; the LHPR denominator is $x + y$.

Task	Canonical primitive sequence and G/P assignment	2L G/P	3L G/P
A	Per source layer: GRASP [G] → PLACE [P].	2/2	3/3
B	Per source block: GRASP [G] → PLACE [P]; two blocks are transferred in every layer.	4/4	6/6
C	Per target: GRASP→EXTRACT [one G] → PLACE [P].	1/1	2/2
D	Per insertion block: GRASP [G] → INSERT [P].	1/1	2/2
E	Each prescribed PUSH is one step; pushes in the first phase occupy G and those in the second phase occupy P.	3/3	6/3
F	Per cycle: GRASP tool [G] → two PUSH steps [G,G] → PLACE tool [P] → two repetitions of GRASP→EXTRACT [G] → PLACE [P].	5/3	10/6

A.4 Stepwise Evaluation Procedure

Evaluation trajectories are scored against the same planned operation sequence using the following procedure:

1. Load the ordered G/P plan associated with the task and layer configuration in Table 4.
2. Mark a step as attempted when its first required primitive begins, and separately determine whether its success criterion is met. Only successfully completed G/P steps enter the LHPR numerator; failed, unstarted, or skipped steps do not. Retries do not create additional progress credit.
3. Mark the step successful only when all primitives assigned to that step meet their completion conditions. A failure of any required primitive makes the parent G/P step unsuccessful without creating an additional step.
4. For each executed step, record the identities of disturbed non-target blocks across all constituent primitives. The same block is counted at most once within that step and is counted again if disturbed during a later step.

5. At episode termination, determine TSR from the complete task state and aggregate successful G steps, successful P steps, executed steps, and disturbed block occurrences according to the metric definitions in Section 3.3.

Task F illustrates the mapping without repeating a separate accounting narrative for every task. In its two-layer configuration, grasping the tool, two tool-assisted pushes, and the two grasp–extract transfers form five G steps. Returning the tool and placing the two extracted blocks form three P steps. The planned sequence is therefore $5\mathrm{G} + 3\mathrm{P}$ and has an LHPR denominator of eight. The three-layer configuration repeats the same collection cycle, yielding $10\mathrm{G} + 6\mathrm{P}$ and a denominator of sixteen. All other task counts follow directly from the corresponding row of Table 4.

B Success, Failure, and Annotation Criteria

B.1 Objective Episode Criteria

We derive all reported metrics from a common set of episode-level event records. Each task is first decomposed into predefined atomic operations, which are assigned to the G and P groups described in Section 3.3. For standard transport tasks, G denotes acquiring or extracting a target block and P denotes placing or inserting it. For pushing and tool-mediated tasks, G and P denote the task-specific operation stages fixed in the corresponding protocol. The operation order and the number of planned G and P steps are fixed before evaluation and are shared by all policies in SIM and REAL.

An episode succeeds only when all prescribed operations are completed and the terminal arrangement meets the task definition: intended blocks occupy their target locations and orientations, the required structure is preserved, and the arrangement remains stable after release. Momentary goal contact followed by displacement or collapse is insufficient. Human review uses the full trajectory to distinguish target contact from stable completion. The operational criteria below specify the placement, restacking, and disturbance thresholds used for annotation, together with the video-based task-completion criteria.

Task-specific scoring criteria. Table 5 summarizes the G/P boundaries, terminal conditions, and observation criteria used for human evaluation.

Table 5: Task-specific human-scoring specification. G/P assignments follow Appendix A.3. Observation criteria apply to the corresponding task-specific operations.

Task	G/P boundaries and terminal success	Pose and observation criteria
A	G: secure acquisition; P: released transfer. All blocks reconstruct the ordered cross tower.	Prescribed orientation judged from video; block placement stable for 10 s.
B	G: acquire each block; P: release at destination. All paired layers preserve spacing and orientation.	Spacing and orientation judged from video; block placement stable for 10 s.
C	G: grasp and fully extract; P: release at tower top. All designated center blocks are restacked.	Restacking height ± 15 mm; stable placement for 10 s; final arrangement judged from video.
D	G: secure acquisition; P: release into slot. All designated insertions completed.	Insertion judged from video; block remains in its prescribed slot for at least 10 s after release.
E	Each prescribed push is one G/P step according to its assigned phase. All ordered pushes completed.	Completion of the prescribed displacement and end state judged from video.
F	G: tool acquisition, each tool push, and each grasp–extraction; P: tool return and block placement. All cycles completed.	Tool return and graspable protrusion judged from video; block placement stable for 10 s.

Disturbance, termination, and validity. For each operation, non-target block-center translation greater than 10 mm counts as disturbance; the fixed foundation and current target are excluded. A block counts once within an operation and again if disturbed in a later operation. Episodes have a 420 s time limit. A policy-induced safety stop or irrecoverable task failure is retained as failure; hardware-failure trials are excluded.

Table 6: Primitive episode records and their use in the benchmark metrics.

Record	Episode-level meaning	Derived use
s_i	Entire task and terminal state satisfy the protocol	TSR
S_i^G	Successfully completed G-group operations	GSR
S_i^P	Successfully completed P-group operations	PSR
C_i	Successful G/P operations, $S_i^G + S_i^P$	LHPR
E_i	Derived counted steps, $C_i + 1 - s_i$	ODR
D_i	Cumulative non-target block-disturbance occurrences	ODR
v_i	Whether the trial is valid and enters the denominator	All metrics

The archived automatic grasp proxy additionally used 0.3 s of prior holding, a 25 mm grasp-point distance, a 70 mm gripper opening, and force/relative-motion checks. These implementation-specific proxies define the archived automatic check; reported episode scores follow the human criteria above.

For episode i, we record terminal success s_i, successful G/P operation counts S_i^G, S_i^P, cumulative non-target disturbance D_i, and validity v_i. Successful progress is $C_i = S_i^G + S_i^P$. The ODR denominator is derived as $E_i = C_i + (1 - s_i)$, not independently annotated. For the disturbance numerator, let A_i denote the number of operations actually started and $\mathcal{B}_{ij}$ the non-target blocks disturbed during operation j:

$$D_i = \sum_{j=1}^{A_i} |\mathcal{B}_{ij}|. \tag{9}$$

A block contributes once per operation and may contribute again in a later operation. The denominator adds one counted step to each unsuccessful episode, including a terminal-state failure. Thus, $\sum_i E_i = \sum_i C_i + N - \sum_i s_i$. This derived count summarizes successful operations and episode outcome; it is distinct from an attempt or retry log. Disturbances from failed operations remain in D_i. Table 6 summarizes the records and their uses.

Metric interpretation. Each planned G/P operation receives success credit at most once. LHPR is the planned-count-weighted combination of GSR and PSR and measures total completed operations. TSR additionally checks the terminal arrangement. Disturbance counts refer to blocks outside the prescribed target set and exclude tools. ODR uses the counted-step denominator defined above; zero denominators and missing annotations produce an unavailable metric.

Among valid episodes, any episode that does not satisfy the complete success predicate is a task failure. A timeout, a policy-induced safety stop, or a policy action that makes further completion impossible remains a valid task failure. A trial is marked invalid only when evaluation integrity is lost for reasons external to the policy, such as hardware or communication failure, missing observations, corrupted records, or an incorrect initial scene. Hardware-failure trials are excluded from the reported cohort; policy-induced failures remain in the denominator. The episode time limit is 420 s.

B.2 Human Annotation and Sampled Agreement

Annotation protocol. Each evaluation episode receives two human annotations, with third-person adjudication of disagreements, using the same criteria in REAL and SIM. Annotators record task success, successful G/P counts, disturbance, and trial validity from the trajectory. Agreed fields are retained; disputed fields are reviewed by a third evaluator against the corresponding video segment. The annotation record associates source ratings and adjudication decisions with the episode identity. All reported metrics are computed from the final human labels in both domains. Inter-annotator agreement is measured on the source ratings before third-person adjudication.

Sampling from source ratings. We assess agreement on a small, stratified subset of Task A, which has explicit G/P operation boundaries. Among 80 eligible episodes per domain with at least two complete source ratings, we select five episodes from each of four policies and two layer settings, yielding 40 REAL and 40 SIM episodes. Sampling uses a single draw without replacement with

seed 42. When more than two ratings exist, the first two annotators in Unicode name order are selected independently of their scores. The sample therefore includes multiple annotator pairs.

Agreement measures. For $q \in \{s, S^G, S^P, \ell\}$, exact agreement is

$$A_q = \frac{1}{N} \sum_{i=1}^{N} \mathbf{1}[q_i^{(1)} = q_i^{(2)}]. \tag{10}$$

Here $\ell_i^{(a)} = (S_i^{G,(a)} + S_i^{P,(a)})/K_i$ is the LHPR assigned by annotator a, with the same planned denominator K_i for both ratings. LHPR agreement is the fraction of episodes with identical normalized progress; G and P agreement is also reported separately.

Table 7: Source-rating agreement on the stratified Task A subset. Agreement uses paired source ratings; annotator pairs vary across episodes.

Field	REAL ($N = 40$)	SIM ($N = 40$)	Combined ($N = 80$)
Task success	100.0%	97.5%	98.75%
Successful G count	100.0%	100.0%	100.0%
Successful P count	100.0%	90.0%	95.0%
LHPR	100.0%	90.0%	95.0%

Final evaluation records. Metrics are computed from the final episode labels obtained through the annotation and adjudication protocol above. The agreement statistics in Table 7 summarize the paired source ratings on the specified Task A sample.

Automatic scoring. The automatic SIM scorer is checked against human-annotated SIM episodes. All reported benchmark results use human labels. The agreement statistics in Table 7 quantify human–human consistency before adjudication; automatic scoring is a separate implementation check.

C Real-World Platform

This section describes the physical hardware, camera arrangement, and task objects used for real-world data collection and evaluation. Figure 15 shows the station and camera arrangement.

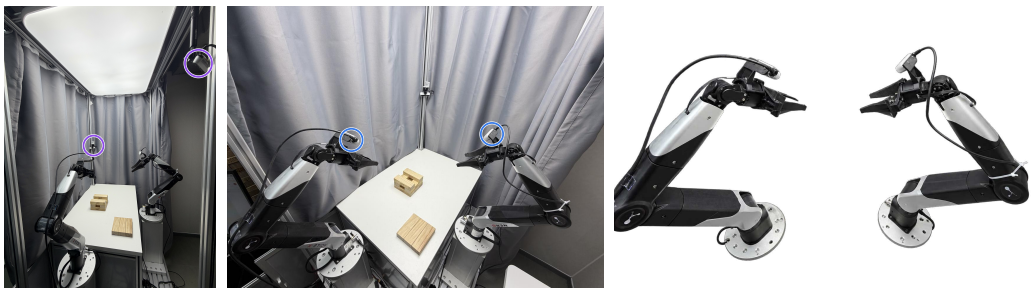


Figure 15: **TOWER real-world platform.** (a) The enclosed station with overhead lighting; violet circles mark the two fixed cameras. (b) The bimanual workspace; blue circles mark the two wrist-mounted cameras. (c) An AI-assisted background-removal illustration of the robot arms, grippers, and wrist cameras. Panels (a) and (b) are original photographs with camera annotations.

(a) Block and foundation (b) Two layers (c) Three layers



Figure 16: **Blocks, foundations, and initial structures.** Crops from recorded REAL frames show a target block, its supporting foundation, and two- and three-layer Task D configurations. Images retain the physical scene; only display rotation and cropping are applied.

C.1 Robot and Control System

The platform uses two seven-degree-of-freedom AGX NERO manipulators, referred to as the left and rear arms. Each arm carries an AGX Gripper. The arms run firmware version 1.20 and communicate over CAN at $1\,\mathrm{Mbps}$. Four Geschwister Schneider CAN–USB adapters provide the leader and follower CAN channels for the two arm systems. Each robot is mounted on a BY-D1200-500 lifting column with an effective travel of $500\,\mathrm{mm}$ and a dedicated BY-D1200 control unit communicating over Modbus TCP.

The PNP-1200-3 station has a $1200 \times 900\,\mathrm{mm}$ chassis and a $960 \times 660\,\mathrm{mm}$ work surface. Its external support frame is approximately $2250\,\mathrm{mm}$ high and carries the lighting and light-shielding enclosure. Illumination is provided by two adjustable three-colour light panels, each approximately $110 \times 70\,\mathrm{cm}$ and rated at $196\,\mathrm{W}$. A GPU workstation handles camera acquisition, policy inference, robot control, and data storage.

C.2 Camera System

Four Intel RealSense D435if RGB-D cameras observe the workspace. Two cameras are mounted on the left and rear wrists, and two are fixed at the left-rear and right-front viewpoints. The RGB streams are configured at 640×480 pixels and $30\,\mathrm{FPS}$. This arrangement provides both stationary workspace views and views that move with the end effectors. Camera mounting hardware includes two PNP-Vision-D435if support kits with extension cables, angle flanges, and adjustable mounts. The four cameras connect through a Genesys Logic USB 3.1 hub. Camera alignment and correspondence with the simulated viewpoints are reported in Appendix E.2.

C.3 Blocks, Foundations, and Initial-State Construction

The real-world platform uses rectangular wooden blocks with nominal dimensions of $150 \times 50 \times 30\,\mathrm{mm}$. Measurements of ten physical specimens yielded lengths of $150.19 \pm 0.25\,\mathrm{mm}$, widths of $50.23 \pm 0.31\,\mathrm{mm}$, and masses of $102.04 \pm 3.37\,\mathrm{g}$; the recorded heights ranged from 29.2 to $30.2\,\mathrm{mm}$. Here, $\pm$ denotes the sample standard deviation across specimens.

Each foundation comprises three side-by-side blocks, forming a nominal $150 \times 150\,\mathrm{mm}$ footprint with a height of $30\,\mathrm{mm}$. Figure 16 illustrates the objects and initial structures. The foundations provide fixed positional references for constructing the task configurations and are distinguished from the blocks manipulated during a trial. Task-specific arrangements include paired, crossed, and aligned upper layers, as well as central gaps for insertion and initially protruding center blocks for extraction.

For real-world initial-state construction, the foundations remain at their designated locations. Initial configurations are varied manually by rotating a set of manipulated blocks relative to these references, while preserving the structure required by each task, including layer arrangements, insertion gaps, and protruding extraction blocks.

D SIMULATION ENVIRONMENT

This section specifies the simulation environment used for evaluation, including scene implementation and rendering, physics and contact parameters, robot control, and initialization. These settings define how policy commands are executed and how observations are generated in simulation.

D.1 SIMULATOR AND SCENE IMPLEMENTATION

Simulation experiments are conducted in NVIDIA Isaac Sim 5.1.0 with GPU-accelerated PhysX dynamics. Images are generated using the RTX real-time renderer in `RaytracedLighting` mode. Each evaluator process instantiates one environment on a selected GPU; parallel evaluations run as independent processes. Figure 17 shows the scene, camera observations, and collision geometry. Scenes are composed in Universal Scene Description (USD), with a task manifest selecting the station, object layout, runtime parameters, and initial robot state.

The station contains two seven-degree-of-freedom Nero manipulators with parallel-jaw grippers. Robot links and task objects use separate visual and collision geometry, with convex collision approximations for robot links and standard blocks. Foundation blocks are kinematic, whereas all manipulated blocks are dynamic rigid bodies. For one taller task-specific support, an override replaces the mesh collider with a cuboid while retaining its visual geometry.

The USD stage is Z-up and uses millimetres (`metersPerUnit`$= 10^{-3}$), while the evaluator interface uses SI units: metres for translation and gripper opening, radians for revolute joints, and seconds for time. Block poses are expressed in the environment frame, and serialized quaternions follow (x, y, z, w) ordering. Four RGB cameras provide right-front, left-back, left-wrist, and back-wrist observations at 640×480 resolution. Rendering is decoupled from physics stepping, so capturing an observation does not advance the simulation.

Rendering and lighting. Four RGB views are rendered at 640×480 pixels using RTX real-time rendering (`RaytracedLighting`) with Deep Learning Anti-Aliasing (DLAA). The direct-lighting, indirect-diffuse, reflection, dome-light, and shadow sampling settings are each set to 16. Regular captures use four render subframes. Motion blur and DLSS frame generation are disabled, while the runtime retains the ray-tracing cache. Image capture does not advance physics.

Lighting combines the station's rectangular overhead light with a runtime dome light. The overhead light has intensity 3000, exposure 0, and enabled colour-temperature adjustment at 5600 K; intensity normalization is disabled. The dome light has intensity 650 and exposure 0. Intensities are reported in native USD light units. These light-exposure values are distinct from camera exposure and renderer post-processing.

D.2 PHYSICS PARAMETERS

Table 8 lists the simulation configuration. We use the Temporal Gauss–Seidel solver with GPU dynamics and GPU broad-phase collision detection.

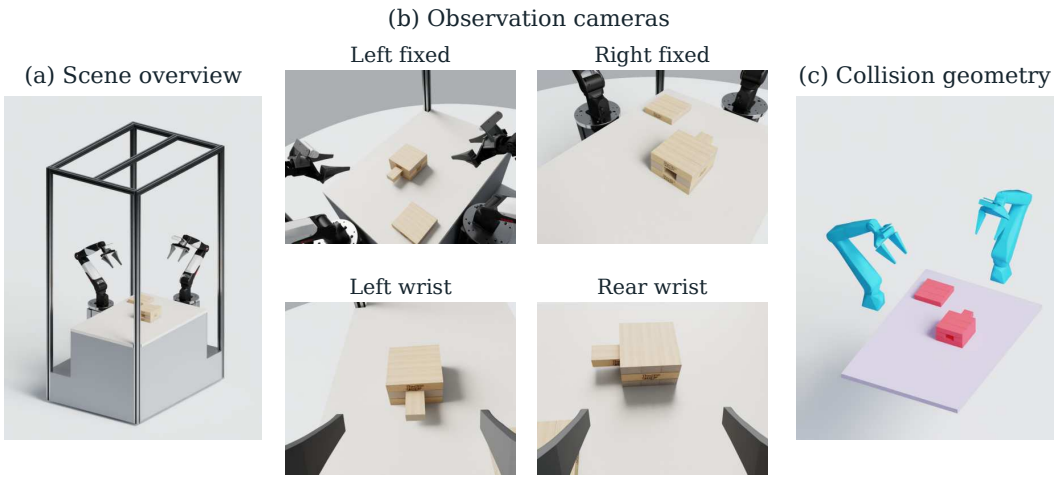


Figure 17: **Simulation scene and observations.** Task C–2 shown through an overview camera, four observation cameras, and a collision-geometry view. Cyan denotes robot geometry, coral denotes blocks, and pale purple denotes the table. Camera mounts are shown as authored input meshes for decomposition.

Table 8: Simulation configuration.

Parameter	Reported configuration
Physics step / simulator step	$1/120$ s / $1/30$ s
Physics substeps per simulator step	4
Gravity	$(0, 0, -9.81)\,\mathrm{m\,s^{-2}}$
Constraint solver	TGS; GPU dynamics and broad phase
Block solver iterations (position / velocity)	$16/4$
Robot solver iterations (position / velocity)	$32/1$
Maximum block depenetration speed	$1\,\mathrm{m\,s^{-1}}$
Block friction (static / dynamic)	$0.551/0.263$
Block restitution	0.02
Finger friction (static / dynamic)	$2.4/2.0$
Standard block size and mass	$0.150 \times 0.050 \times 0.030\,\mathrm{m}$; $0.1056\,\mathrm{kg}$
Taller task-specific support size and mass	$0.150 \times 0.050 \times 0.032\,\mathrm{m}$; $0.11264\,\mathrm{kg}$
Finger effort limit	3 N per finger
Initial settling duration	1 s

Robot-link inertial properties are inherited from the composed USD assets. The implementation supports arm-specific drive-gain, armature, and joint-friction bundles with runtime readback checks. Their correspondence with measured robot motion is evaluated separately in Appendix E.4. The block masses above are assigned at runtime; the taller support mass is scaled in proportion to its height.

The final SIM evaluations use block coefficients $0.551/0.263$, matching the force-gauge estimates in Appendix E.3.

D.3 Robot Controller and Action Execution

The archived simulator interface accepts a 16-dimensional absolute-position command comprising seven joint targets and one gripper-opening target for each arm. Joint targets are expressed in radians and gripper openings in metres. Commands are mapped to named articulation degrees of freedom and applied through position drives; the non-commanded lifter joints remain fixed in this implementation. This rollout path uses joint-space commands directly and does not invoke inverse kinematics or an operational-space controller.

SIM evaluation applies the fitted robot-dynamics parameters and arm delays.

Action channels and execution. The $\pi_{0.5}$ adapter receives 18 absolute physical channels, ordered as seven joints, lift, and gripper for each arm. It selects zero-based indices $(0{:}6, 8, 9{:}15, 17)$ for the simulator and records but discards lift targets at indices 7 and 16. Simulator observations insert logical zeros in those two lift slots; the physical lifters remain at their configured poses. Both lifts remain fixed for all four SIM policies.

SIM and REAL execute the first 16 predicted targets before requesting new observations and another prediction (Section 4.1).

Gripper openings are clipped to $[0, 0.1]\,\mathrm{m}$ and converted to equal and opposite finger displacements. Commands with an invalid dimension or a non-finite entry are rejected. Arm targets are not additionally clipped in command space; the articulation's joint, velocity, and effort limits remain active. At reset, the evaluator restores object poses, clears dynamic-object velocities, initializes robot state and position targets, and allows the scene to settle for 1 s. It then restores the task-specific robot state with zero joint velocity and clears the episode clock and controller history before rollout.

D.4 Simulation Initialization and Randomization

Each environment is initialized from a scene manifest and a saved robot configuration. Reset restores object poses and clears dynamic-body velocities. Following 1 s of physical settling, robot joint positions are restored with zero joint velocities, and the simulation clock and controller history are reset.

Layout generation is reproducible through a seed derived from the base scene seed and a layout identifier. The reported evaluations reuse fixed layout identifiers and robot initializations across repetitions; scene perturbations are not resampled, and optional asset randomization is disabled. Repetitions therefore share the configured layout and robot initialization.

E Real-to-Sim Alignment

This section describes how the simulated environment is aligned with the physical setup through scene reconstruction, camera alignment, contact and friction measurements, and robot-response calibration. We report each component's measurement procedure, fitted parameters, and validation results.

E.1 Scene Reconstruction

Workstation. The physical workstation was reconstructed through an agent-assisted CAD-to-USD workflow. The station CAD assembly provided the geometric basis for the simulated work surface and supporting structure, while photographs of the physical setup guided the refinement of visible materials and appearance. The station was assembled with the two robot models and task objects in a common simulation scene. Figure 18(a) shows the resulting workstation using the station asset referenced by the local $\pi_{0.5}$ evaluation configuration.

Jenga blocks. The block asset combines a supplied mesh with photo-guided surface reconstruction. The mesh was cleaned and normalized to nominal dimensions of $150 \times 50 \times 30\,\mathrm{mm}$. Photographs guided the generation of face-specific textures representing wood grain and printed markings; roughness and normal maps provide additional surface detail. Thus, the asset combines explicit geometry with photograph-conditioned appearance, rather than constituting a photogrammetric reconstruction of an individual specimen. Figure 18(b) shows the resulting simulation asset.

Measured object variability. The physical blocks are not dimensionally identical. Measurements of ten specimens yielded lengths of $150.19 \pm 0.25\,\mathrm{mm}$ and widths of $50.23 \pm 0.31\,\mathrm{mm}$, while observed heights ranged from 29.2 to $30.2\,\mathrm{mm}$. Their masses were $102.04 \pm 3.37\,\mathrm{g}$. Here, $\pm$ denotes the sample standard deviation across specimens. These measurements characterize the physical

(a) Simulated workstation



(b) Simulated Jenga block



Figure 18: **Scene reconstruction.** Reconstructed assets for the simulation environment: (a) the simulated workstation and (b) the simulated Jenga block.

objects and distinguish their variability from the nominal geometry of the canonical simulation asset. Simulation mass and contact parameters are reported separately with physics alignment.

E.2 Camera Alignment

Static-camera data and conventions. We align the static RGB-D camera poses with the metric station model without using fiducial tags as optimization evidence. Each camera contributes 20 synchronized 640×480 RGB frames with depth aligned to the color imager. Frames 0–14 are used for pose selection and frames 15–19 are held out until the pose is frozen. Intrinsics are taken from the recorded camera profiles and remain fixed. The cameras are physically inverted; the 180° rotation is applied only for visualization and is not included in either the intrinsic or extrinsic matrices. The right-front camera is the optimized view, whereas the left-back camera is evaluated as a verification-only view and is not re-optimized on the final capture.

Static-camera optimization. The calibration scene contains the $960 \times 660\,\mathrm{mm}$ worktop and 41 measured Jenga-block placements distributed over three heights. A robust table-plane estimate initializes roll, pitch, and camera height. For the right-front camera, a Sobol search evaluates 8,192 hypotheses over yaw and the optical-axis intersection with the worktop. Independent pose basins are then refined using coarse-to-fine point-to-plane ICP on tabletop and fixed hardware points. Occlusion-aware Jenga silhouettes disambiguate symmetric tabletop solutions. The selected basin undergoes a final six-degree-of-freedom refinement using equally normalized table and Jenga silhouette, contour, and depth residuals. Pixels lacking simulated support receive a fixed penalty rather than being discarded. Movable robot pixels are excluded from the objective and are used only for qualitative full-frame inspection.

Static-camera held-out evaluation. Table 9 reports evaluation after freezing the camera poses. Depth entries are the median, 75th-percentile, and 90th-percentile absolute errors over fixed real pixels.

Table 9: Held-out static-camera alignment on frames 15–19. Depth errors are in millimetres.

Camera role	Jenga IoU	Table P50/P75/P90	Jenga P50/P75/P90
Right-front (optimized)	0.9353	2.39/3.71/6.60	3.59/8.63/15.91
Left-back (verification)	0.9095	5.11/10.68/22.13	7.92/14.44/24.49

The right-front result is the calibrated static-camera extrinsic. The left-back row quantifies cross-view verification rather than an independently optimized final pose. Remaining disagreement can

arise from robot-state timing, station-asset geometry, illumination, and material appearance; these effects are not absorbed into the camera transform.

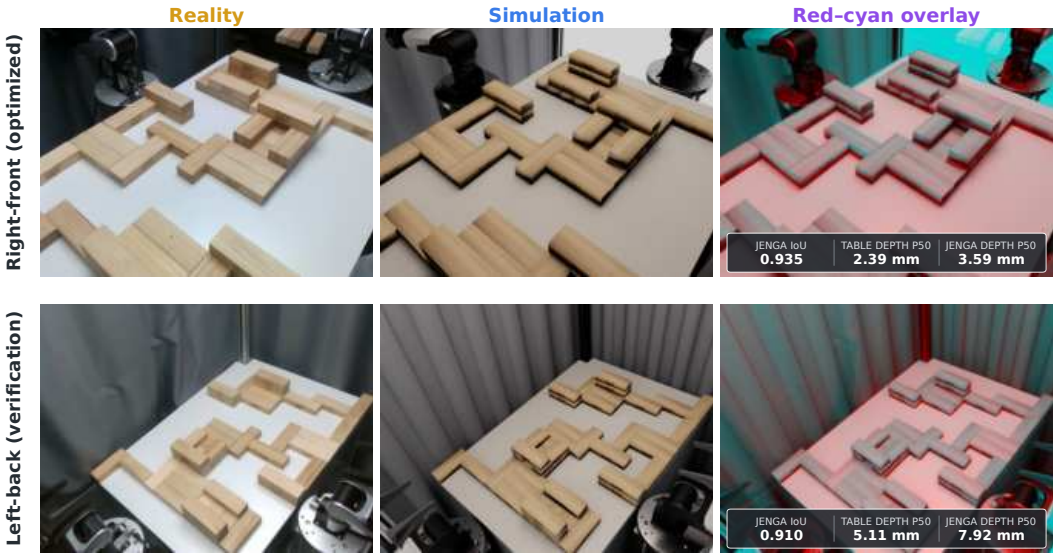


Figure 19: Qualitative validation of the frozen static-camera extrinsics. Each row shows the recorded image, the simulation rendered from the corresponding camera, and a red–cyan overlay (red: reality; cyan: simulation). The right-front camera is optimized, whereas the left-back camera is a verification-only view. Images are rotated 180° for display; the calibration remains in the original optical frame.

Wrist camera-to-gripper calibration. Each wrist camera is calibrated as a rigid camera-to-gripper transform using manually reviewed silhouettes of the two gripper fingers. We sample four temporally separated frames (0, 1600, 3200, and 4787) from a 4,788-frame episode. Fingers are annotated separately, while the camera mount, gripper base, robot body, blocks, table, and background are excluded. Let $\boldsymbol{\xi} \in \mathbb{R}^6$ be a translation and axis–angle residual applied to the nominal transform, such that $T(\boldsymbol{\xi}) = \Delta T(\boldsymbol{\xi}) T_0$. Isaac Sim renders per-link instance masks at each recorded arm configuration. For frame k, we minimize

$$\mathcal{L}_k = \left(1 - \overline{\mathrm{IoU}}_k\right) + 0.25\left(1 - \mathrm{IoU}_k^{\cup}\right) + 0.05\, d_k^{\mathrm{edge}}, \tag{11}$$

where the first term averages the two finger IoUs, the second measures their union, and d^{edge} is the mean symmetric edge Chamfer distance in pixels. The frame-average objective includes 0.01 times a bound-normalized quadratic prior. Fixed-seed differential evolution followed by Powell refinement searches within $\pm 25\,\mathrm{mm}$ and $\pm 12^\circ$.

Wrist-camera selection and validation. For the left wrist, calibration raises the mean per-finger IoU from 0.727 to 0.855 and reduces median edge error from 7.83 to 4.23 pixels. Fits from separate episode splits agree within $1.37\,\mathrm{mm}$ and 0.65°. For the back wrist, the nominal transform already achieves 0.912 mean IoU and a 2.35-pixel median edge error; the optimized candidate is worse and inconsistent across splits. We therefore accept the optimized left-wrist residual but retain a zero residual for the back wrist. Table 10 summarizes the quantitative evidence for these choices.

Table 10: Wrist-camera model selection. IoU is mean per-finger intersection over union (higher is better); edge error is the median symmetric Chamfer distance in pixels (lower is better).

Camera	Nominal IoU	Optimized IoU	Nominal edge	Optimized edge	Retained
Left wrist	0.727	**0.855**	7.83	**4.23**	Optimized
Back wrist	**0.912**	0.856	**2.35**	3.98	Nominal

For completeness, the retained residual vectors are

$$\boldsymbol{\xi}_{\text{left}} = (16.46, -10.54, -2.55;\ -4.71, -7.14, 1.34), \qquad \boldsymbol{\xi}_{\text{back}} = \mathbf{0}, \tag{12}$$

where the first three entries are translation in millimetres and the final three are axis–angle rotation in degrees.

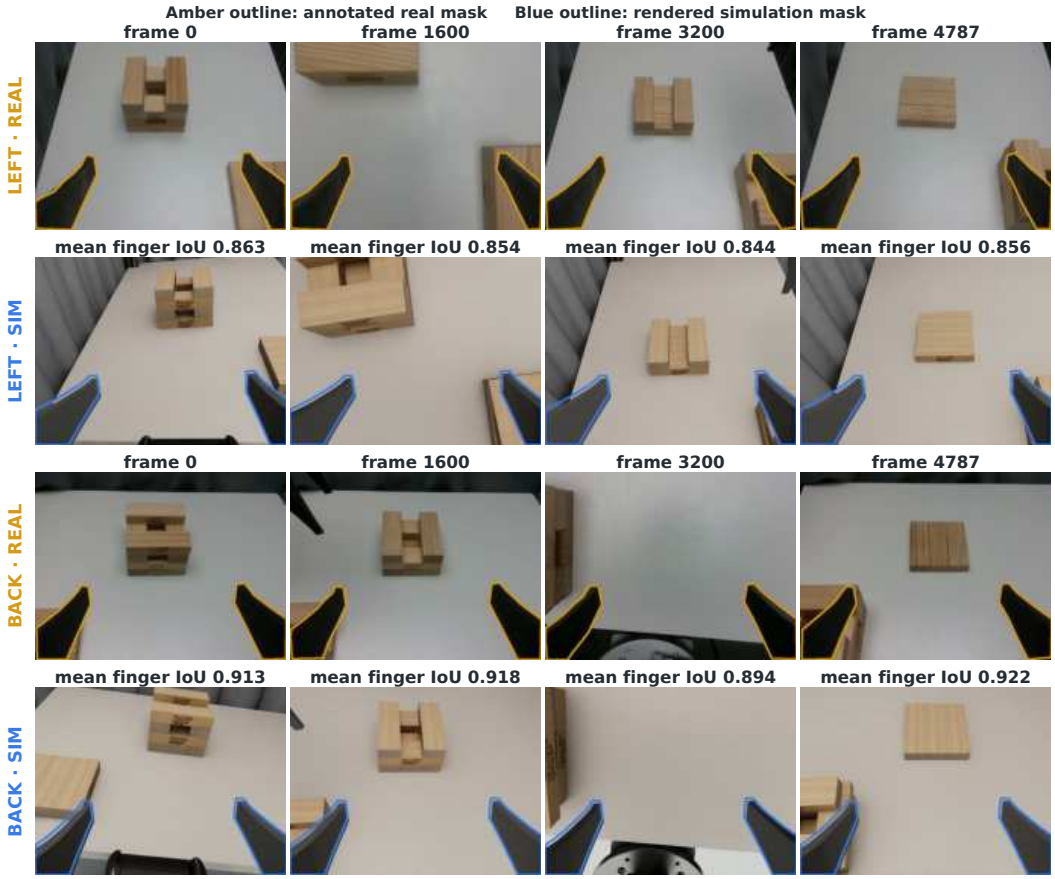
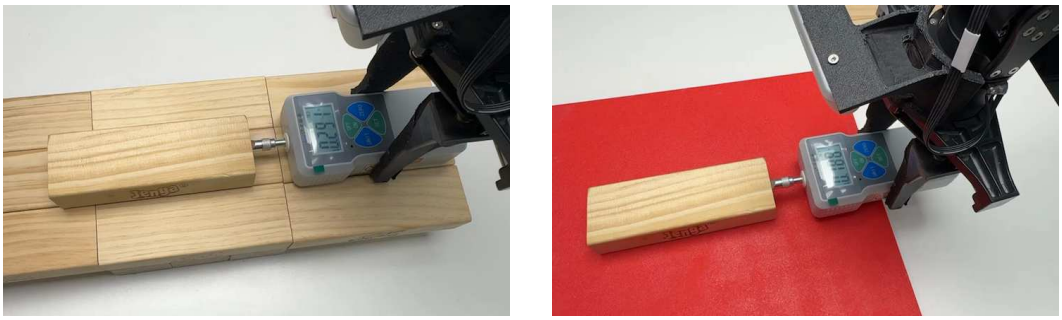


Figure 20: Accepted wrist-camera silhouette alignment at all four sampled poses. The optimized left-wrist transform remains consistent across the episode, while the back wrist retains its nominal transform. Amber outlines show the annotated real masks and blue outlines show the rendered simulation masks.

The final transforms are checked over the complete 4,788-frame replay and at all four sampled poses. The calibration aligns gripper silhouettes across views and provides a visual-proxy estimate of the wrist-camera transform.

E.3 Physics Alignment

Apparatus and protocol. We measured Jenga–Jenga and Jenga–PLA contact using a NERO robot arm, a digital force gauge, and a single $96.6\,\mathrm{g}$ test block. The block was placed on a level Jenga or PLA support and loaded only by its own weight, giving a normal force of $F_n = mg = 0.9473\,\mathrm{N}$. The robot translated the gauge horizontally against the block. We performed ten static and ten dynamic trials for each contact pair. The static measurement was the maximum force immediately before breakaway. For the dynamic measurement, the force-gauge display was sampled from $30\,\mathrm{fps}$ video at $2\,\mathrm{Hz}$, perspective-rectified, and manually reviewed. Unreadable frames and frames outside the sliding interval were excluded. For each trial, we averaged the readings from 0.5 to $1.5\,\mathrm{s}$ after force first exceeded $0.1\,\mathrm{N}$, thereby excluding the initial loaded sample and the stationary holding tail.



(a) Jenga–Jenga contact (b) Jenga–PLA contact

Figure 21: Contact-friction measurement setup. The robot-held force gauge contacts the test block on (a) a Jenga surface and (b) a PLA surface.

Coefficient estimates. For each trial, we computed $\mu = F_t/F_n$, where F_t is the measured tangential force. Table 11 reports the mean and sample standard deviation across the ten trials.

Table 11: Measured friction coefficients (mean $\pm$ sample SD; ten trials per contact pair and regime).

Contact pair	Static coefficient	Dynamic coefficient
Jenga–Jenga	0.551 ± 0.046	0.263 ± 0.029
Jenga–PLA	0.392 ± 0.073	0.242 ± 0.027

Simulator material parameters. Under the average material-combination rule, $\mu_{AB} = (\mu_A + \mu_B)/2$, the measured Jenga–Jenga coefficient sets the Jenga material value. Solving $\mu_{\text{PLA}} = 2\mu_{\text{Jenga,PLA}} - \mu_{\text{Jenga}}$ yields $(\mu_s, \mu_d) = (0.551, 0.263)$ for Jenga and $(0.234, 0.221)$ for PLA. The PLA values are inverse-fit settings for the measured cross-material contact, distinct from a PLA–PLA measurement. Dynamic coefficients are estimated from the sliding-force window; velocity and gauge response delay were not synchronously recorded. The measured Jenga pair was applied in the final SIM evaluations. Result-linked deployment records document the Jenga pair; the inverse-fit PLA values and average combination rule are reported as proposed settings (Appendix D.2).

E.4 Robot-Response Calibration

Scope and data collection. We align each NERO arm independently to free-motion joint trajectories. The procedure identifies effective closed-loop response parameters for simulation. For each of the seven joints, we recorded small- and medium-step motions and slow- and fast-sine motions. This produced 28 fitting and 28 disjoint held-out trajectories per arm. Each recording contains commanded and measured joint positions, measured motor velocity, and command and feedback timestamps.

Replay and fitting objective. Commands are replayed in Isaac Lab using zero-order hold. Simulated states are interpolated to the original, nonuniform feedback timestamps before comparison. Only the intentionally excited joint contributes to a trajectory's loss. For trajectory r, the objective is

$$\mathcal{L}_r(\theta) = \text{MSE}(q_{\text{sim}} - q_{\text{real}}) + 0.05\ \text{MSE}(\dot{q}_{\text{sim}} - \dot{q}_{\text{real}})\,. \tag{13}$$

Losses are first averaged within each motion type and then equally across the four types, preventing the longer sine trajectories from dominating the step trajectories. Identification uses a robot-only scene; the second arm, table contacts, and Jenga objects are excluded.

Optimization. We use two-stage CMA-ES with population eight, 12 generations per stage, and seed zero. Stage 1 fits seven armatures, seven viscous-friction coefficients, seven Coulomb-friction coefficients, and one shared command delay per arm. The respective search ranges are $[10^{-5}, 1]\,\text{kg}\,\text{m}^2$, $[0, 7]$, $[0, 0.5]$, and zero to five control steps. Stage 2 freezes these values and fits

seven K_p and seven K_d gains in logarithmic space, with ranges $[1, 10^5]$ and $[0.01, 4000]$, respectively. The fitted shared delays are $5\,\mathrm{ms}$ for the left arm and $10\,\mathrm{ms}$ for the back arm. The fitted arm delays are applied during SIM evaluation.

Held-out validation. Table 12 compares the source-scene baseline with the fitted model on the 28 held-out trajectories per arm. The fit reduces aggregate position and velocity errors in both arms; per-joint validation is summarized below.

Table 12: **Robot-response validation.** Aggregate held-out trajectory error; lower is better.

Arm	Metric	Baseline	Fitted	Improvement
Left	Position RMSE (deg)	1.693	**1.218**	**28.0%**
	Velocity RMSE ($\mathrm{rad\,s^{-1}}$)	0.507	**0.278**	**45.2%**
Back	Position RMSE (deg)	1.705	**1.390**	**18.5%**
	Velocity RMSE ($\mathrm{rad\,s^{-1}}$)	0.508	**0.248**	**51.1%**

Held-out response results. The fitted model reduces held-out position RMSE by 28.0% and 18.5%, and velocity RMSE by 45.2% and 51.1%, for the left and back arms, respectively. Per-joint checks identify a left joint 7 friction value of 6.943, within one percent of its search bound, and back-arm position-error increases of 26.2% and 19.1% for joints 4 and 7. The latter exceed the five-percent degradation criterion, so the fit retains provisional status under the validation protocol. Multiple-seed fits yield similar trajectory errors with different gain settings. These results characterize the effective free-motion response model used for calibration.

F Dataset and Demonstration Collection

F.1 Demonstration Collection

REAL demonstrations are collected through leader–follower teleoperation on the platform in Appendix C. SIM demonstrations are generated by scripted task controllers in the environment in Appendix D. Both follow the task-specific initial states and G/P operation sequences in Appendix A. Each episode contains one task and layer setting and records four camera streams, robot state, command targets, end-effector poses, and timestamps. Corresponding camera views are matched by physical viewing direction. Hardware and scene illustrations are provided in the platform and simulation appendices and are not repeated here.

The demonstration dataset spans 455.7 hours across six tasks, two tower heights, and two domains. Each of the 24 domain–task–layer settings provides 500 training and two validation episodes: 12,000 training and 48 validation episodes in total (6,000 training and 24 validation episodes per domain). The 960 policy-evaluation episodes are reported separately from these demonstrations.

Original recordings are retained. Demonstration selection screens recording integrity and suitability for training; it is distinct from evaluation validity. Policy-induced failures remain valid evaluation outcomes under Appendix B.

F.2 Unified Data Representation

REAL and SIM trajectories use a shared physical representation, summarized in Table 13. The 18-dimensional vector is ordered as left-arm J1–J7, left lift J8, left gripper, followed by the corresponding channels of the back arm. Original HDF5 recordings store the fourteen arm joints and two grippers separately from the two lift channels; conversion assembles them into the shared vector.

Table 13: Shared trajectory representation. Training samples are constructed at 10 Hz; source recordings retain their original timestamps.

Modality	Representation	Semantics and use
Images	Four RGB streams	Two global and two wrist views; policy observations.
Robot state	18 dimensions	Measured arm angles (rad), lift heights (mm), and absolute gripper widths.
Action	18 dimensions	Recorded command targets; model-specific relative transforms are applied during policy adaptation.
End-effector pose	14 dimensions	Position and quaternion (x, y, z, w) per arm in its base frame; retained for auditing, not the joint-policy action target.
Task instruction	One string per episode	Shared instruction for corresponding SIM and REAL task settings.
Timing	Frame indices and timestamps	Preserve observation–command alignment and source-frame correspondence.
SIM annotations	Object states and events	Support automatic evaluation; excluded from shared policy inputs.

Recorded lift state is the measured height; the demonstration interface stores lift command targets on the $\{0, 50, \ldots, 500\}$ mm grid. This recording convention is distinct from a policy's continuous output representation. Absolute physical quantities are retained in the source data. Action horizons, relative-action transforms, and normalization are specified per model rather than encoded as a universal policy target. The complete storage schema is retained in the accompanying implementation documentation.

F.3 Data Construction and Split

Curated episodes are ordered by collection start time within each domain–task–layer setting. Of the 502 demonstrations in each setting, the first two are held out for validation and the remaining 500 supply training data. Held-out episodes do not contribute to normalization statistics. The chronological holdout supports offline data and model checks; closed-loop performance is measured through the evaluation rollouts. Model-specific internal splits can further reduce the effective training subset.

Training samples are converted at 10 Hz with common sampling indices across modalities, preserving observation–command correspondence and source-frame indices. Post-conversion checks verify dimensions, episode counts, instructions, and video readability. The shared representation is separated from each policy's action transform and normalization, which are applied during model-specific adaptation.

G Complete Evaluation Protocol

G.1 Training–Evaluation Domain Matrix

The reported comparison evaluates REAL-trained policies in REAL and SIM-trained policies in SIM. Each architecture is trained separately for each task–layer configuration and domain (Section 4.1). Matching is defined at the configuration level through shared architectures, task definitions, and scoring criteria. Each pair compares domain-specific checkpoints and aggregates independently collected episodes within the corresponding configuration. REAL demonstrations are collected through teleoperation, and SIM demonstrations are generated by scripted controllers (Appendix F.1).

G.2 Trial Counts and Metric Implementation

Reported cohort. The quantitative comparison covers Tasks A–F at two and three layers, with four policies and ten valid episodes per domain and configuration. This gives 120 episodes per policy and domain, 480 per domain, and 960 in total. Invalid-trial criteria follow Appendix B.1.

Paper-compatible aggregation reference. The accompanying `reproduction/paper_metrics` package implements TSR, GSR, PSR, successful-operation LHPR, ODR, pooled model ODR, Pearson correlation, normalized MMRV, and CCC from episode labels. The reference implementation checks incomplete labels, duplicate identities, group counts, aggregation, and tie handling. It is separate from automatic trajectory scoring.

Configuration-level aggregation. The operation definitions and planned G/P counts are given in Appendix A, and primitive annotation fields in Table 6. For each policy–task–layer–domain group, the computation proceeds as follows:

1. Retain valid episodes and verify that success labels, successful G/P counts, disturbance counts, and planned counts are available. Missing annotations are not replaced with zero; an affected metric remains unavailable until its required fields are resolved.
2. Derive successful progress $C_i = S_i^G + S_i^P$ and counted steps $E_i = C_i + 1 - s_i$ for each episode, using the convention in Section 3.3. Failed attempts do not earn progress credit, and E_i is not a measured attempt count.
3. Compute TSR from the mean binary success label. Sum successful and planned G/P counts before dividing to obtain GSR, PSR, and LHPR; sum disturbances and counted steps before dividing to obtain ODR. Use the equations in Section 3.3. A zero denominator yields an undefined metric rather than an imputed value.

Success and progress rates are displayed as percentages; ODR is reported as block-disturbance occurrences per conventionally counted step, without percentage scaling. Mean successful progress, $\bar{C}$, retains units of operations per episode and is distinct from LHPR.

Model-level and cross-domain summaries. For model-level ODR, pool disturbance counts and counted steps over the same twelve configurations before division; do not average configuration-level ratios. Lower pooled ODR ranks higher. For the correlation analysis, compute one mean successful-step count per model and configuration in each domain, then pair the domain means. Appendix G.3 defines the resulting association and ranking statistics. All summaries use the same twelve task–layer configurations.

G.3 Statistical Analysis

Matched observations. The cohort specified in Appendix G.2 yields 48 paired model–configuration means. Each pair comprises the SIM and REAL mean for the same model and task–layer configuration.

For model m, task–layer configuration t, and domain d, let $C_i^d = S_i^{G,d} + S_i^{P,d}$ be the number of successfully completed semantic operations. We compare $\bar{C}_{m,t}^d = n^{-1} \sum_i C_i^d$ between domains. With planned length $K_t = N_t^G + N_t^P$, normalized progress is $L_{m,t}^d = \bar{C}_{m,t}^d / K_t$. Raw progress expresses domain differences in successful semantic operations per episode, while LHPR expresses them relative to the prescribed task length. Failed attempts receive no progress credit; complete G/P progress can still violate the terminal-state criterion. The aggregate count measures completed planned operations across the episode.

Association and ranking agreement. For a set of matched model–configuration pairs $\mathcal{I}$, write $x_i = \bar{C}_{m,t}^{\mathrm{SIM}}$ and $y_i = \bar{C}_{m,t}^{\mathrm{REAL}}$. We compute Pearson correlation on these mean successful step counts:

$$r_{\mathcal{I}} = \frac{\sum_{i \in \mathcal{I}} (x_i - \bar{x})(y_i - \bar{y})}{\sqrt{\sum_{i \in \mathcal{I}} (x_i - \bar{x})^2 \sum_{i \in \mathcal{I}} (y_i - \bar{y})^2}}. \tag{14}$$

Pearson summarizes linear association in execution progress. We complement it with normalized mean maximum rank violation (MMRV), following the directed comparison rule of Li et al. (2025). Let $R_{m,t} = \bar{C}^{\mathrm{REAL}}_{m,t}/K_t$ and $S_{m,t} = \bar{C}^{\mathrm{SIM}}_{m,t}/K_t$. For $M = 4$ policies and the shared set $\mathcal{T}$ of twelve task–layer configurations,

$$V_{m,t} = \max_{j \neq m} |R_{m,t} - R_{j,t}| \, \mathbf{1}\big[(S_{m,t} > S_{j,t}) \neq (R_{m,t} > R_{j,t})\big], \tag{15}$$

$$\mathrm{MMRV}_{\mathcal{T}} = \frac{1}{|\mathcal{T}|} \sum_{t \in \mathcal{T}} \frac{1}{M} \sum_{m=1}^{M} V_{m,t}. \tag{16}$$

Each policy contributes its largest real-performance margin associated with an ordering disagreement; a policy with no such disagreement contributes zero. Strict comparisons retain the directed treatment of ties: a simulated tie can disagree with a strict real ordering. We first compare policies within each configuration and then average configurations equally. Dividing by K_t before aggregation makes MMRV dimensionless and bounded by $[0, 1]$, avoiding larger margins solely due to longer planned sequences. Scatter coordinates and Pearson remain in raw steps; normalization changes the MMRV scale, not the observations or within-configuration ordering.

Absolute error and interpretation. The mean absolute error is $|\mathcal{I}|^{-1} \sum_{i \in \mathcal{I}} |x_i - y_i|$, expressed in successful operations per episode. The main scatter and Pearson use raw step counts, whereas MMRV uses task-normalized progress. We report these as complementary descriptive summaries over the fixed set of 48 model–configuration pairs: raw-step association, absolute progress differences, and normalized policy-ordering disagreement.

Normalized agreement and CCC. For the same 48 pairs, set $u_i = x_i/K_{t(i)}$ and $v_i = y_i/K_{t(i)}$. For normalized analysis, Pearson correlation is computed on (u_i, v_i), and $100|\mathcal{I}|^{-1} \sum_i |u_i - v_i|$ gives MAE in percentage points. The concordance correlation coefficient (CCC) is

$$\rho_c = \frac{2\sigma_{uv}}{\sigma_u^2 + \sigma_v^2 + (\bar{u} - \bar{v})^2}, \qquad \sigma_{uv} = \frac{1}{|\mathcal{I}|} \sum_i (u_i - \bar{u})(v_i - \bar{v}), \tag{17}$$

where both variances also use divisor $|\mathcal{I}|$. Unlike Pearson correlation alone, CCC penalizes differences in location and scale as well as imperfect association. Normalization retains the same 48 configuration-mean pairs and scales each pair by its prescribed task length. A zero variance makes Pearson undefined, and a zero CCC denominator is also reported as undefined rather than imputed.

G.4 Complete Metrics and Episode-Level Variability

Tables 14 and 15 provide the complete five-metric results for the two- and three-layer configurations, supporting the task-performance analysis in Section 4.3. Tables 16 and 17 report the mean and sample standard deviation of the five recorded fields for each policy–task–layer configuration in each domain. For ten episodes, $\bar{x} = \frac{1}{10} \sum_{i=1}^{10} x_i$ and $\mathrm{SD}(x) = \sqrt{\frac{1}{9} \sum_{i=1}^{10} (x_i - \bar{x})^2}$. These statistics describe variation across episodes using the original binary success labels and operation or disturbance counts.

Table 14: Complete policy performance: Two-layer configurations. Each task column pairs SIM (top) and REAL (bottom). TSR, GSR, PSR, and LHPR are percentages (↑); ODR is disturbed block-occurrences per counted step (↓).

Policy	Metric	A SIM / REAL	B SIM / REAL	C SIM / REAL	D SIM / REAL	E SIM / REAL	F SIM / REAL
$\pi_{0.5}$	TSR ↑	80.0 100.0	70.0 80.0	100.0 100.0	0.0 10.0	0.0 10.0	0.0 0.0
	GSR ↑	95.0 100.0	87.5 90.0	100.0 100.0	100.0 100.0	20.0 50.0	36.0 44.0
	PSR ↑	90.0 100.0	85.0 90.0	100.0 100.0	0.0 30.0	83.3 80.0	0.0 0.0
	LHPR ↑	92.5 100.0	86.2 90.0	100.0 100.0	50.0 65.0	51.7 65.0	22.5 27.5
	ODR ↓	0.05 0.00	0.07 0.05	0.50 0.50	0.00 1.05	0.61 0.21	1.50 1.34
StarVLA-α	TSR ↑	40.0 20.0	90.0 70.0	100.0 90.0	50.0 0.0	60.0 10.0	60.0 100.0
	GSR ↑	95.0 95.0	100.0 87.5	100.0 100.0	100.0 100.0	86.7 56.7	100.0 100.0
	PSR ↑	70.0 60.0	97.5 87.5	100.0 90.0	70.0 0.0	86.7 90.0	100.0 100.0
	LHPR ↑	82.5 77.5	98.8 87.5	100.0 95.0	85.0 50.0	86.7 73.3	100.0 100.0
	ODR ↓	0.00 0.03	0.00 0.11	0.30 0.50	1.36 0.00	0.11 0.15	0.00 0.00
OpenWAM	TSR ↑	100.0 0.0	100.0 100.0	10.0 60.0	40.0 20.0	0.0 90.0	20.0 80.0
	GSR ↑	100.0 100.0	100.0 100.0	100.0 100.0	100.0 100.0	26.7 96.7	90.0 100.0
	PSR ↑	100.0 50.0	100.0 100.0	10.0 80.0	40.0 40.0	96.7 100.0	80.0 100.0
	LHPR ↑	100.0 75.0	100.0 100.0	55.0 90.0	70.0 70.0	61.7 98.3	86.2 100.0
	ODR ↓	0.00 0.00	0.00 0.04	2.50 0.18	1.40 1.05	0.09 0.05	0.34 0.21
Fast-WAM	TSR ↑	100.0 100.0	100.0 100.0	0.0 100.0	0.0 0.0	0.0 60.0	20.0 70.0
	GSR ↑	100.0 100.0	100.0 100.0	90.0 100.0	40.0 100.0	36.7 93.3	86.0 96.0
	PSR ↑	100.0 100.0	100.0 100.0	0.0 100.0	0.0 0.0	46.7 93.3	83.3 90.0
	LHPR ↑	100.0 100.0	100.0 100.0	45.0 100.0	20.0 50.0	41.7 93.3	85.0 93.8
	ODR ↓	0.00 0.00	0.00 0.00	1.79 0.50	0.71 1.00	0.86 0.12	0.09 0.09

TSR: task success rate; GSR: planned G-operation success rate; PSR: planned P-operation success rate; LHPR: successful operations divided by planned operations; ODR: cumulative disturbance divided by counted steps. Each configuration uses 10 episodes per domain. Ratios are computed after pooling their numerators and denominators within each configuration. Counted steps follow $E_i = C_i + 1 - s_i$. Planned G/P counts and operation definitions follow Appendix A.

Table 15: Complete policy performance: Three-layer configurations. Each task column pairs SIM (top) and REAL (bottom). TSR, GSR, PSR, and LHPR are percentages (↑); ODR is disturbed block-occurrences per counted step (↓).

Policy	Metric	A SIM / REAL	B SIM / REAL	C SIM / REAL	D SIM / REAL	E SIM / REAL	F SIM / REAL
$\pi_{0.5}$	TSR ↑	70.0	0.0	0.0	0.0	0.0	0.0
		80.0	10.0	40.0	0.0	0.0	0.0
	GSR ↑	96.7	21.7	70.0	50.0	25.0	16.0
		100.0	38.3	90.0	95.0	58.3	21.0
	PSR ↑	86.7	18.3	50.0	0.0	10.0	5.0
		93.3	30.0	65.0	5.0	20.0	5.0
	LHPR ↑	91.7	20.0	60.0	25.0	20.0	11.9
		96.7	34.2	77.5	50.0	45.6	15.0
	ODR ↓	0.19	1.06	1.82	3.50	1.96	1.45
		0.02	0.12	0.68	2.00	1.00	1.29
StarVLA-α	TSR ↑	100.0	10.0	20.0	0.0	10.0	0.0
		90.0	70.0	30.0	0.0	0.0	10.0
	GSR ↑	100.0	66.7	100.0	100.0	58.3	54.0
		100.0	90.0	100.0	90.0	36.7	57.0
	PSR ↑	100.0	63.3	55.0	45.0	56.7	40.0
		96.7	85.0	65.0	10.0	3.3	41.7
	LHPR ↑	100.0	65.0	77.5	72.5	57.8	48.8
		98.3	87.5	82.5	50.0	25.6	51.2
	ODR ↓	0.00	0.09	1.97	0.74	0.51	0.70
		0.07	0.10	0.55	2.30	2.52	0.67
OpenWAM	TSR ↑	70.0	100.0	70.0	10.0	0.0	20.0
		90.0	30.0	90.0	0.0	50.0	30.0
	GSR ↑	93.3	100.0	100.0	95.0	48.3	74.0
		93.3	83.3	100.0	80.0	90.0	73.0
	PSR ↑	80.0	100.0	90.0	50.0	0.0	66.7
		90.0	81.7	100.0	30.0	96.7	63.3
	LHPR ↑	86.7	100.0	95.0	72.5	32.2	71.2
		91.7	82.5	100.0	55.0	92.2	69.4
	ODR ↓	0.13	0.00	1.00	1.11	0.62	0.39
		0.11	0.08	0.24	2.81	0.11	0.39
Fast-WAM	TSR ↑	40.0	0.0	0.0	0.0	0.0	20.0
		100.0	90.0	30.0	0.0	0.0	30.0
	GSR ↑	83.3	63.3	50.0	55.0	23.3	63.0
		100.0	95.0	90.0	100.0	21.7	62.0
	PSR ↑	63.3	53.3	50.0	5.0	0.0	55.0
		100.0	95.0	85.0	55.0	0.0	53.3
	LHPR ↑	73.3	58.3	50.0	30.0	15.6	60.0
		100.0	95.0	87.5	77.5	14.4	58.8
	ODR ↓	0.14	0.21	0.67	2.73	0.04	0.98
		0.02	0.03	0.57	1.29	5.00	0.94

TSR: task success rate; GSR: planned G-operation success rate; PSR: planned P-operation success rate; LHPR: successful operations divided by planned operations; ODR: cumulative disturbance divided by counted steps. Each configuration uses 10 episodes per domain. Ratios are computed after pooling their numerators and denominators within each configuration. Counted steps follow $E_i = C_i + 1 - s_i$. Planned G/P counts and operation definitions follow Appendix A.

Table 16: Episode-level means and standard deviations: Two-layer configurations. Each cell reports mean ± sample standard deviation, with SIM above REAL ($n = 10$ per domain and configuration).

Policy	Field	A SIM / REAL	B SIM / REAL	C SIM / REAL	D SIM / REAL	E SIM / REAL	F SIM / REAL
$\pi_{0.5}$	s	0.80 ±0.42 1.00 ±0.00	0.70 ±0.48 0.80 ±0.42	1.00 ±0.00 1.00 ±0.00	0.00 ±0.00 0.10 ±0.32	0.00 ±0.00 0.10 ±0.32	0.00 ±0.00 0.00 ±0.00
	g	1.90 ±0.32 2.00 ±0.00	3.50 ±0.97 3.60 ±0.97	1.00 ±0.00 1.00 ±0.00	1.00 ±0.00 1.00 ±0.00	0.60 ±0.70 1.50 ±1.27	1.80 ±0.79 2.20 ±0.63
	p	1.80 ±0.42 2.00 ±0.00	3.40 ±1.07 3.60 ±0.97	1.00 ±0.00 1.00 ±0.00	0.00 ±0.00 0.30 ±0.48	2.50 ±0.53 2.40 ±0.70	0.00 ±0.00 0.00 ±0.00
	C	3.70 ±0.67 4.00 ±0.00	6.90 ±2.02 7.20 ±1.93	2.00 ±0.00 2.00 ±0.00	1.00 ±0.00 1.30 ±0.48	3.10 ±0.99 3.90 ±1.29	1.80 ±0.79 2.20 ±0.63
	D	0.20 ±0.42 0.00 ±0.00	0.50 ±0.71 0.40 ±0.52	1.00 ±0.00 1.00 ±0.00	0.00 ±0.00 2.30 ±1.64	2.50 ±1.58 1.00 ±0.82	4.20 ±1.14 4.30 ±1.49
StarVLA-α	s	0.40 ±0.52 0.20 ±0.42	0.90 ±0.32 0.70 ±0.48	1.00 ±0.00 0.90 ±0.32	0.50 ±0.53 0.00 ±0.00	0.60 ±0.52 0.10 ±0.32	0.60 ±0.52 1.00 ±0.00
	g	1.90 ±0.32 1.90 ±0.32	4.00 ±0.00 3.50 ±0.85	1.00 ±0.00 1.00 ±0.00	1.00 ±0.00 1.00 ±0.00	2.60 ±0.70 1.70 ±0.95	5.00 ±0.00 5.00 ±0.00
	p	1.40 ±0.52 1.20 ±0.42	3.90 ±0.32 3.50 ±0.85	1.00 ±0.00 0.90 ±0.32	0.70 ±0.48 0.00 ±0.00	2.60 ±0.70 2.70 ±0.48	3.00 ±0.00 3.00 ±0.00
	C	3.30 ±0.67 3.10 ±0.57	7.90 ±0.32 7.00 ±1.70	2.00 ±0.00 1.90 ±0.32	1.70 ±0.48 1.00 ±0.00	5.20 ±1.32 4.40 ±0.84	8.00 ±0.00 8.00 ±0.00
	D	0.00 ±0.00 0.10 ±0.32	0.00 ±0.00 0.80 ±0.63	0.60 ±0.84 1.00 ±0.00	3.00 ±2.58 0.00 ±0.00	0.60 ±0.84 0.80 ±0.79	0.00 ±0.00 0.00 ±0.00
OpenWAM	s	1.00 ±0.00 0.00 ±0.00	1.00 ±0.00 1.00 ±0.00	0.10 ±0.32 0.60 ±0.52	0.40 ±0.52 0.20 ±0.42	0.00 ±0.00 0.90 ±0.32	0.20 ±0.42 0.80 ±0.42
	g	2.00 ±0.00 2.00 ±0.00	4.00 ±0.00 4.00 ±0.00	1.00 ±0.00 1.00 ±0.00	1.00 ±0.00 1.00 ±0.00	0.80 ±0.42 2.90 ±0.32	4.50 ±0.53 5.00 ±0.00
	p	2.00 ±0.00 1.00 ±0.00	4.00 ±0.00 4.00 ±0.00	0.10 ±0.32 0.80 ±0.42	0.40 ±0.52 0.40 ±0.52	2.90 ±0.32 3.00 ±0.00	2.40 ±0.52 3.00 ±0.00
	C	4.00 ±0.00 3.00 ±0.00	8.00 ±0.00 8.00 ±0.00	1.10 ±0.32 1.80 ±0.42	1.40 ±0.52 1.40 ±0.52	3.70 ±0.67 5.90 ±0.32	6.90 ±0.74 8.00 ±0.00
	D	0.00 ±0.00 0.00 ±0.00	0.00 ±0.00 0.30 ±0.95	5.00 ±0.00 0.40 ±0.52	2.80 ±2.30 2.30 ±2.06	0.40 ±1.26 0.30 ±0.67	2.60 ±2.37 1.70 ±2.21
Fast-WAM	s	1.00 ±0.00 1.00 ±0.00	1.00 ±0.00 1.00 ±0.00	0.00 ±0.00 1.00 ±0.00	0.00 ±0.00 0.00 ±0.00	0.00 ±0.00 0.60 ±0.52	0.20 ±0.42 0.70 ±0.48
	g	2.00 ±0.00 2.00 ±0.00	4.00 ±0.00 4.00 ±0.00	0.90 ±0.32 1.00 ±0.00	0.40 ±0.52 1.00 ±0.00	1.10 ±0.88 2.80 ±0.42	4.30 ±0.67 4.80 ±0.63
	p	2.00 ±0.00 2.00 ±0.00	4.00 ±0.00 4.00 ±0.00	0.00 ±0.00 1.00 ±0.00	0.00 ±0.00 0.00 ±0.00	1.40 ±0.52 2.80 ±0.42	2.50 ±0.97 2.70 ±0.95
	C	4.00 ±0.00 4.00 ±0.00	8.00 ±0.00 8.00 ±0.00	0.90 ±0.32 2.00 ±0.00	0.40 ±0.52 1.00 ±0.00	2.50 ±1.18 5.60 ±0.52	6.80 ±1.55 7.50 ±1.58
	D	0.00 ±0.00 0.00 ±0.00	0.00 ±0.00 0.00 ±0.00	3.40 ±1.07 1.00 ±0.00	1.00 ±0.00 2.00 ±2.11	3.00 ±1.49 0.70 ±0.67	0.70 ±0.95 0.70 ±0.95

s: episode-level task success (1 for success, 0 for failure); g: number of successful grasp operations; p: number of successful placement operations; $C = g + p$: total successful operation steps; D: cumulative number of disturbed non-target block-occurrences, counting repeated disturbance across operations. Each entry shows the mean ± sample standard deviation across 10 episodes; the standard deviation uses divisor $n - 1 = 9$. Values are original binary labels or counts, not percentages; D is a disturbance count, not ODR.

Table 17: Episode-level means and standard deviations: Three-layer configurations. Each cell reports mean ± sample standard deviation, with SIM above REAL ($n = 10$ per domain and configuration).

Policy	Field	A SIM / REAL	B SIM / REAL	C SIM / REAL	D SIM / REAL	E SIM / REAL	F SIM / REAL
$\pi_{0.5}$	s	0.70 ±0.48 0.80 ±0.42	0.00 ±0.00 0.10 ±0.32	0.00 ±0.00 0.40 ±0.52	0.00 ±0.00 0.00 ±0.00	0.00 ±0.00 0.00 ±0.00	0.00 ±0.00 0.00 ±0.00
	g	2.90 ±0.32 3.00 ±0.00	1.30 ±0.67 2.30 ±2.21	1.40 ±0.52 1.80 ±0.42	1.00 ±0.00 1.90 ±0.32	1.50 ±0.85 3.50 ±0.85	1.60 ±1.17 2.10 ±2.23
	p	2.60 ±0.70 2.80 ±0.42	1.10 ±0.32 1.80 ±1.81	1.00 ±0.00 1.30 ±0.67	0.00 ±0.00 0.10 ±0.32	0.30 ±0.95 0.60 ±0.84	0.30 ±0.95 0.30 ±0.95
	C	5.50 ±0.97 5.80 ±0.42	2.40 ±0.97 4.10 ±3.73	2.40 ±0.52 3.10 ±0.99	1.00 ±0.00 2.00 ±0.47	1.80 ±1.14 4.10 ±1.60	1.90 ±1.97 2.40 ±3.13
	D	1.10 ±0.74 0.10 ±0.32	3.60 ±2.76 0.60 ±0.97	6.20 ±3.46 2.50 ±2.27	7.00 ±0.00 6.00 ±3.02	5.50 ±2.64 5.10 ±1.73	4.20 ±5.27 4.40 ±5.78
StarVLA-α	s	1.00 ±0.00 0.90 ±0.32	0.10 ±0.32 0.70 ±0.48	0.20 ±0.42 0.30 ±0.48	0.00 ±0.00 0.00 ±0.00	0.10 ±0.32 0.00 ±0.00	0.00 ±0.00 0.10 ±0.32
	g	3.00 ±0.00 3.00 ±0.00	4.00 ±1.41 5.40 ±1.26	2.00 ±0.00 2.00 ±0.00	2.00 ±0.00 1.80 ±0.42	3.50 ±1.18 2.20 ±1.69	5.40 ±3.20 5.70 ±3.13
	p	3.00 ±0.00 2.90 ±0.32	3.80 ±1.40 5.10 ±1.45	1.10 ±0.57 1.30 ±0.48	0.90 ±0.32 0.20 ±0.42	1.70 ±0.82 0.10 ±0.32	2.40 ±1.96 2.50 ±1.90
	C	6.00 ±0.00 5.90 ±0.32	7.80 ±2.78 10.50 ±2.55	3.10 ±0.57 3.30 ±0.48	2.90 ±0.32 2.00 ±0.67	5.20 ±1.75 2.30 ±1.83	7.80 ±5.05 8.20 ±4.96
	D	0.00 ±0.00 0.40 ±0.52	0.80 ±1.23 1.10 ±1.37	7.70 ±2.71 2.20 ±1.48	2.90 ±3.90 6.90 ±0.74	3.10 ±2.47 8.30 ±4.06	6.20 ±2.53 6.10 ±2.33
OpenWAM	s	0.70 ±0.48 0.90 ±0.32	1.00 ±0.00 0.30 ±0.48	0.70 ±0.48 0.90 ±0.32	0.10 ±0.32 0.00 ±0.00	0.00 ±0.00 0.50 ±0.53	0.20 ±0.42 0.30 ±0.48
	g	2.80 ±0.63 2.80 ±0.63	6.00 ±0.00 5.00 ±1.05	2.00 ±0.00 2.00 ±0.00	1.90 ±0.32 1.60 ±0.52	2.90 ±0.32 5.40 ±0.70	7.40 ±2.01 7.30 ±2.06
	p	2.40 ±0.97 2.70 ±0.95	6.00 ±0.00 4.90 ±0.99	1.80 ±0.42 2.00 ±0.00	1.00 ±0.47 0.60 ±0.84	0.00 ±0.00 2.90 ±0.32	4.00 ±1.63 3.80 ±1.69
	C	5.20 ±1.40 5.50 ±1.58	12.00 ±0.00 9.90 ±2.02	3.80 ±0.42 4.00 ±0.00	2.90 ±0.74 2.20 ±1.23	2.90 ±0.32 8.30 ±0.82	11.40 ±3.53 11.10 ±3.73
	D	0.70 ±0.95 0.60 ±0.70	0.00 ±0.00 0.90 ±1.60	4.10 ±0.88 1.00 ±0.00	4.20 ±3.01 9.00 ±3.02	2.40 ±1.96 1.00 ±0.94	4.80 ±1.62 4.60 ±2.37
Fast-WAM	s	0.40 ±0.52 1.00 ±0.00	0.00 ±0.00 0.90 ±0.32	0.00 ±0.00 0.30 ±0.48	0.00 ±0.00 0.00 ±0.00	0.00 ±0.00 0.00 ±0.00	0.20 ±0.42 0.30 ±0.48
	g	2.50 ±0.53 3.00 ±0.00	3.80 ±0.79 5.70 ±0.95	1.00 ±0.00 1.80 ±0.42	1.10 ±0.32 2.00 ±0.00	1.40 ±0.52 1.30 ±1.95	6.30 ±2.67 6.20 ±2.90
	p	1.90 ±0.99 3.00 ±0.00	3.20 ±0.79 5.70 ±0.95	1.00 ±0.00 1.70 ±0.48	0.10 ±0.32 1.10 ±0.32	0.00 ±0.00 0.00 ±0.00	3.30 ±1.95 3.20 ±2.10
	C	4.40 ±1.51 6.00 ±0.00	7.00 ±1.49 11.40 ±1.90	2.00 ±0.00 3.50 ±0.85	1.20 ±0.63 3.10 ±0.32	1.40 ±0.52 1.30 ±1.95	9.60 ±4.50 9.40 ±4.97
	D	0.70 ±0.82 0.10 ±0.32	1.70 ±1.06 0.30 ±0.95	2.00 ±1.76 2.40 ±1.71	6.00 ±0.94 5.30 ±1.64	0.10 ±0.32 11.50 ±4.62	10.20 ±4.29 9.50 ±4.60

s: episode-level task success (1 for success, 0 for failure); g: number of successful grasp operations; p: number of successful placement operations; $C = g + p$: total successful operation steps; D: cumulative number of disturbed non-target block-occurrences, counting repeated disturbance across operations. Each entry shows the mean ± sample standard deviation across 10 episodes; the standard deviation uses divisor $n - 1 = 9$. Values are original binary labels or counts, not percentages; D is a disturbance count, not ODR.

H Failure Analysis

H.1 Screening and Failure Definitions

Evidence and assignment unit. The failure analysis assigns one provisional primary visible mode to each screened episode using sampled-frame overviews and selected temporal windows. Task-failure episodes enter mode screening; successful episodes retain their success status. Assignment follows the visible interaction stage and the existing screening note, with task-specific mappings

listed in Table 18. This mode-screening procedure is distinct from the double-annotation and adjudication protocol for the episode scores.

Table 18: Operational mapping of the six provisional modes displayed in Figure 5. "Explicit" denotes an action stage identified in the screening note; "task-inferred" denotes a stage assigned using the task-specific mapping.

Displayed mode	Visible criterion	Stage evidence and permitted inference
Insertion failure	The target remains misaligned, protruding, or outside its required slot.	An insertion/slot/push-in note is explicit; a generic placement-or-insertion note is mapped to insertion only for Task D.
Push-induced instability	Non-target blocks move or the structure tilts or separates during direct pushing.	A direct-push note is explicit; a generic contact-instability note is mapped to direct pushing only for Task E.
Placement/release failure	A transferred target does not remain at the required destination pose.	Placement/release notes are explicit; generic placement-or-insertion notes outside Task D receive a task-inferred placement/release stage.
Extraction-induced disturbance	Neighboring blocks move during target extraction.	Requires an extraction-stage note attached to a structural-disturbance label; an extraction shortfall alone is insufficient.
Block-grasp failure	The attempt fails to establish stable possession of the block.	Requires a block-grasp-stage note; tool grasp and unresolved grasp-versus-holding labels remain separate.
Tool-push instability	The structure shifts or destabilizes during tool contact or pushing.	Requires a tool-push/contact note attached to a structural-disturbance label; the task name alone is insufficient.

H.2 Selection and Quantitative Results

Selection and denominators. The six displayed categories are the most frequent mapped action-stage–mode combinations after excluding unknown types, unlocalized stages, and unresolved compound stages. Selection uses pooled REAL and SIM counts, not the size of the cross-domain gap. The two largest categories are tied. The selected categories contain 204 records (94 REAL and 110 SIM), comprising 103 stage-explicit and 101 task-inferred assignments. Remaining records include scored successes, unresolved labels, and other mapped categories; they are not reassigned to the displayed modes and remain in the denominator. Each plotted percentage is the category count divided by all 480 episodes in its domain, including successful episodes. The six categories form a subset of the cohort, so their percentages need not sum to 100%.

Synchronization with final scores. Screening labels are joined to the final evaluation records by policy, domain, and episode source path. The failure-only filter excludes nine previously screened SIM records whose final task-success labels are positive, including one extraction-disturbance case. Thirteen retained screened records have updated score fields, seven within the displayed categories. Their existing stage labels are carried forward in this score-filter update; new failure modes are not inferred from G/P counts. Frequencies use the retained provisional mode labels and all 480 records per domain as the denominator.

Label conventions. Composite or unlocalized stage notes retain their existing labels. Each episode contributes at most one screened primary mode; co-occurring phenomena remain in the source notes. Representative frames illustrate these labels, while category frequencies are computed from episode records. Matched-task examples compare visible outcomes across domains and identify the policy used in each case.

H.3 Representative Cases

Figure 5 pairs REAL and SIM frames from matching tasks at comparable interaction stages. Frames are rotated, cropped, and uniformly scaled for presentation; they are not additional observations. Extraction and grasping examples include different policies across domains; the panels compare task-matched visible outcomes.

Structural evolution across operations. The examples connect operation outcomes to the changing block arrangement. Pushing and extraction can alter neighbouring contacts and support; release transfers support from the gripper to the tower. Subsequent insertion or placement must accommodate the resulting geometry. This interpretation links the visible cases to the long-horizon discussion in Section 4.2; the mode counts summarize the corresponding visible outcomes.

Figure 22 expands the examples to three REAL/SIM pairs per mode. Each image identifies its task, layer count, and policy. All pairs share a task family; the third grasp pair compares two- and three-layer settings. Images come from distinct episode records and show selected states within the annotated event windows.

Table 19: Six retained provisional modes. Percentages use all 480 evaluation records per domain; categories are not exhaustive.

Provisional mode	REAL, count (%)	SIM, count (%)
Insertion failure	26 (5.42)	39 (8.13)
Push-induced instability	31 (6.46)	34 (7.08)
Placement/release failure	20 (4.17)	11 (2.29)
Extraction-induced disturbance	7 (1.46)	9 (1.88)
Block-grasp failure	4 (0.83)	10 (2.08)
Tool-push instability	6 (1.25)	7 (1.46)

Insertion and direct-push instability are the two largest displayed categories in both domains. The selected SIM records contain 39 insertion cases and 11 placement/release cases; the corresponding REAL counts are 26 and 20. The labels describe operation outcomes and non-target structural changes. The tabulated frequencies use the same domain denominators as Figure 5.

I Reproducibility and Evaluation Scope

This section specifies the distributed materials and evaluation scope.

I.1 Reproducibility

The public review materials at `https://anonymous.4open.science/r/artifact-3d0e01332a1edd93` provide simulator interfaces, policy-serving adapters, scene/asset manifests, and documentation. The release contains code and templates; evaluated weights and episode records are separate artifacts. Its scored transfer runner covers Task B, with four- and six-block transfers for the two tower heights.

The release runner and paper aggregation reference serve distinct roles. The runner records completed-prefix progress and unique disturbed blocks per episode. The paper reference computes successful-operation LHPR and operation-specific ODR from human episode labels. For the reported cohort, the evaluation time limit is 420 s and completion is assessed using the human criteria in Appendix B.1. These settings are specified independently of release defaults and proposed automatic-check options.

The paper aggregation reference is pinned by version and SHA256 in its manifest. It validates episode identities and labels, then exports metric tables, agreement statistics, annotation-status summaries, and a result-to-artifact mapping. The accompanying CSV defines the input schema. Task-specific G/P plans and human-scoring instructions cover Tasks A–F; the reference operates on those labels.

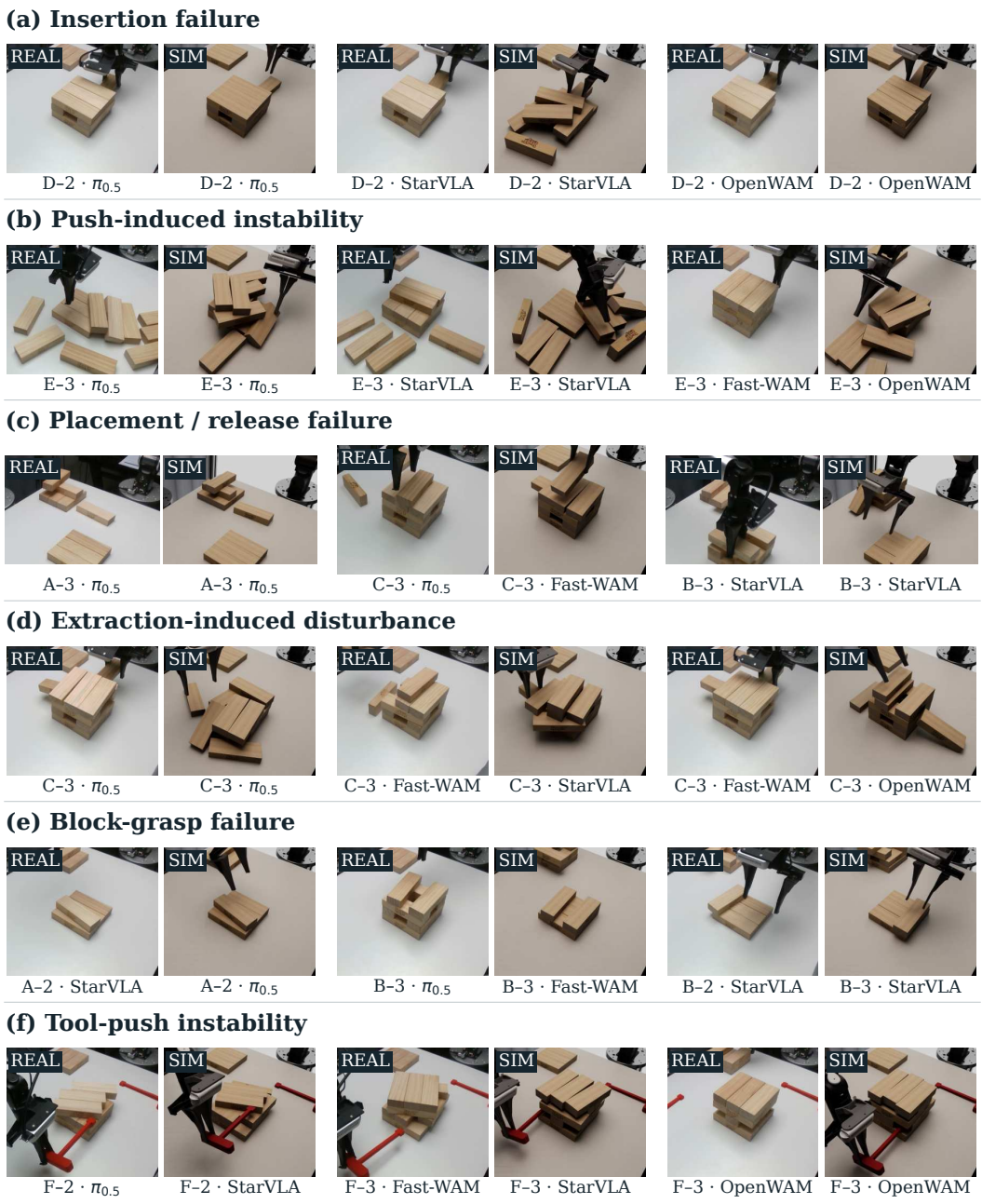


Figure 22: **Additional REAL/SIM failure examples.** Each row represents one provisional mode and contains three pairs (REAL left, SIM right). Labels identify task, layers, and policy; some pairs use different policies, and the third grasp pair uses different layer counts. These are qualitative examples from the evaluation cohort, not additional counted trials.

A reproducible evaluation associates each policy–task–layer–domain group with its checkpoint, normalization and preprocessing metadata, run configuration, episode identities, and final annotations. These artifacts connect policy execution to the paper's aggregation procedure.

Simulation runs on an RTX 5000 PRO GPU. Training uses multiple PPU-ZW810E accelerators, each with 96 GiB of device memory.

I.2 Evaluation Scope

The evaluation covers six rigid-block task families at two tower heights, under the specified lighting and table appearance. Four policy architectures are trained and evaluated within each domain. Shared task definitions and scoring criteria support configuration-level comparisons of completion, progress, disturbance, and failure patterns.