Robotics: Science and Systems 2026
Sydney, Australia, July 13–17, 2026

RoboLab: A High-Fidelity Simulation Benchmark for Analysis of Task Generalist Policies

Xuning Yang[1], Rishit Dagli[2,4], Alex Zook[1], Hugo Hadfield[1],
Ankit Goyal[1], Stan Birchfield[1], Fabio Ramos[1,3], and Jonathan Tremblay[1]
[1]NVIDIA, [2]University of Toronto, [3]The University of Sydney, [4]Work done during internship at NVIDIA

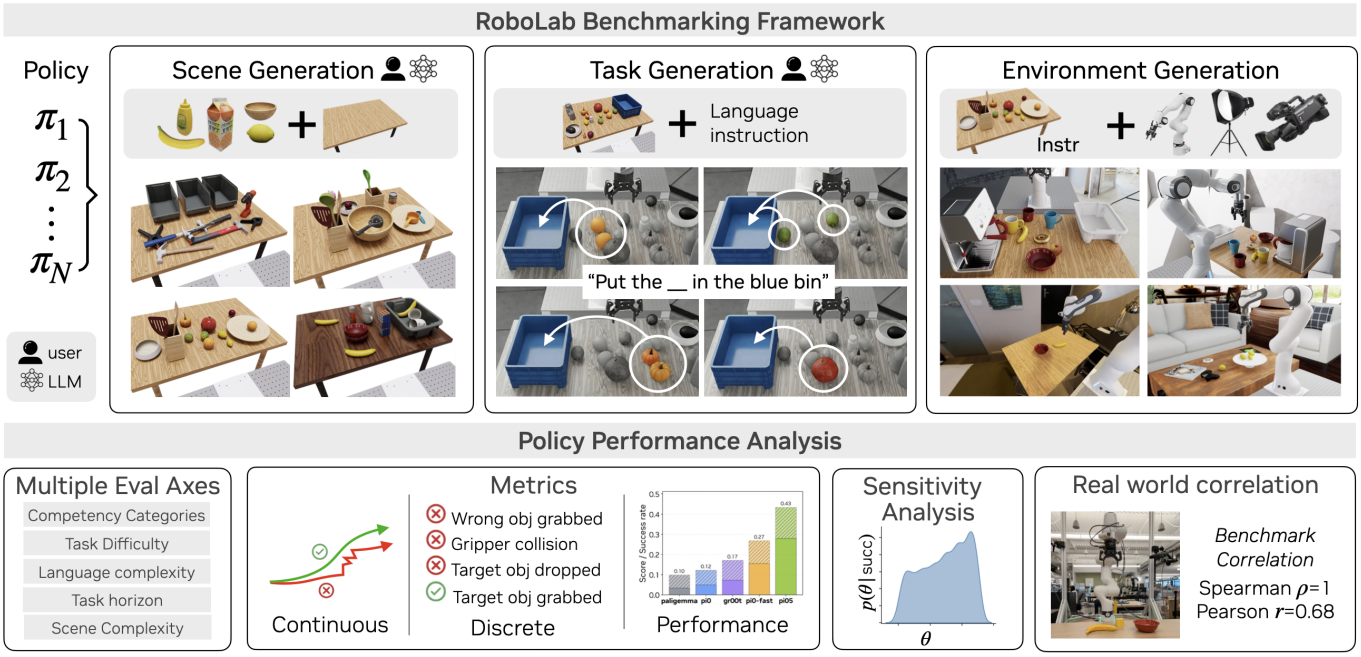


Fig. 1: **Overview of RoboLab.** RoboLab addresses large-scale evaluation of real-world policies via a novel evaluation framework. By introducing a streamlined pipeline for generating new scenes and tasks (top row), RoboLab enables rapid extensibility for testing generalization capabilities. The accompanying benchmark, *RoboLab-120*, introduces various evaluation axes and a suite of metrics and analysis tools, and demonstrate benchmark-level correlation with real-world benchmarks (bottom row).

***Abstract*—The pursuit of general-purpose robotics has yielded impressive foundation models, yet simulation-based benchmarking remains a bottleneck due to rapid performance saturation and a lack of true generalization testing. Existing benchmarks often exhibit significant domain overlap between training and evaluation, trivializing success rates and obscuring insights into robustness. We introduce RoboLab, a simulation benchmarking framework designed to address these challenges. Concretely, our framework is designed to answer two questions: (1) to what extent can we understand the performance of a real-world policy by analyzing its behavior in simulation, and (2) which factor most strongly affect policy behavior. First, RoboLab enables human-authored and LLM-enabled generation of scenes and tasks in a robot- and policy-agnostic manner within a high-fidelity simulation environment. We introduce an accompanying RoboLab-120 benchmark, consisting of 120 tasks categorized into three competency axes: visual, procedural, relational, across three difficulty levels. Second, we introduce a systematic analysis of real-world policies that quantify both their performance and the sensitivity of their behavior to controlled perturbations, exposing significant performance gap in current state-of-the-art models. By providing granular metrics and a scalable toolset, RoboLab offers a scalable framework for evaluating the true generalization capabilities of task-generalist robotic policies. Project website: https://research.nvidia.com/labs/srl/projects/robolab/.**

I. Introduction

The pursuit of generality has been a longstanding challenge in modern robotics. Recent advances have produced impressive generalist robot policies that demonstrate success in challenging and novel tasks in the real-world. Despite this progress, benchmarks for evaluating whether these policies are truly task-general has been slow. Evaluating models in the real world remains prohibitively expensive and logistically expensive, motivating the rise of simulation-based benchmarks as an appealing alternative.

Current robotics benchmarks [19, 36, 11, 16] face several critical limitations: (1) a lack of high-fidelity simulation aimed at evaluation of real-world policies; (2) rapid performance saturation on static task sets; and (3) a lack of granular analysis regarding policy failure modes.

For instance, popular benchmarks like LIBERO [19] often utilize nearly identical environments for both training and

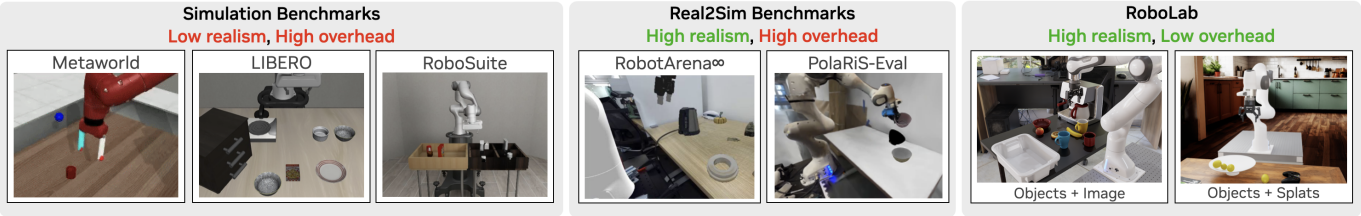


Fig. 2: Three approaches for robotic benchmarks. LEFT: To date, pure simulation based benchmarks have exhibited low visual quality, creating a large sim2real transfer gap. MIDDLE: Real2sim benchmarks address this issue by using techniques to bring real-world visual texture into simulation. However, these environments are extremely costly with reported per-scene generation time of ∼1hr [10]. RIGHT: Our approach achieves a high degree of realism with low overhead.

evaluation. When policies are fine-tuned on these simulation-specific demonstrations, the lack of a meaningful task domain gap trivializes the evaluation process and obscures the model's true generalization capabilities. Many existing platforms have limited realism or are difficult to extend due to using rigid architectures that make it cumbersome to introduce new objects, tasks, or robots (Fig. 2).

To address these limitations, we present RoboLab (Fig. 1), a benchmarking framework designed for rigorous evaluation of generalist policies trained on real-world data. Unlike prior benchmarks that rely on PDDL or rigid scene-graph definitions [19], RoboLab introduces an easy-to-use framework that enables human-authored and AI-enabled scene and task generation. This enables fast creation of hundreds of new tasks and scenes, providing a scalable framework that mitigates benchmark saturation and ensures long-term value.

We introduce the RoboLab-120 benchmark, comprising of 120 hand-curated diverse pick-and-place tasks, aimed at evaluating the generalization capabilities. These tasks span varying difficulties (65 simple, 38 moderate, 18 complex) and multiple competency axes (44 relational, 91 visual, and 36 procedural). This benchmark reflects tasks encountered in "in the wild" household scenarios. To prevent evaluation of policies overfitted to a particular simulation domain, we evaluate policies trained exclusively on the real-world DROID [13] dataset. Our analysis shows that the results on RoboLab-120 achieve benchmark-level correlation with real-world benchmarks [1], indicating that RoboLab is a meaningful proxy for real-world evaluation for task-generalism.

Lastly, we introduce a suite of analysis tools that gives insight into the model performance beyond binary success rates and broader understanding of policy performance; including subtask scoring, event tracking, Bayesian-based sensivity analysis of performance given scene variations via Neural Posterior Estimation.

In summary, our contributions are:

1) **RoboLab**: A novel benchmarking platform designed for evaluating real-world robotics policies with a scalable, AI-based workflow capable of procedurally generating over hundreds of unique robot- and policy- agnostic scenes and tasks, built on IsaacLab[29].
2) **RoboLab-120 Benchmark**: 120 tasks diversely represented across three distinct competency axes (visual, procedural, relational), across 3 levels of difficulty, and supported by robustness metrics.
3) **Policy Analysis**: We introduce a suite of analysis tools that gives insight into the model performance beyond binary success rates and broader understanding of policy performance.

II. RELATED WORK

Simulation-Based Benchmarks. Simulation provides a scalable and reproducible environment for evaluating robot manipulation policies. Widely used benchmarks such as RLBench [11], MetaWorld [34], and robosuite [37], ManiSkill2 [7], CALVIN [20], LIBERO [19], and BEHAVIOR-1K [15], offer standardized task suites for learning and evaluation in simulation across pre-defined task families and object configurations. However, in these settings, policies are typically trained and evaluated in the same simulated environments, which encourages overfitting to simulator-specific quirks, leads to rapid benchmark saturation, and makes real-world generalization hard to assess. [36]. In our setting, policies are instead trained on large-scale real-world data (e.g., DROID [13]), while high-fidelity simulation is used only as a controlled evaluation environment. Training and evaluation domains are decoupled and measured performance more closely reflects robustness in the real world. Some benchmarks also solely focus on perturbations and variations to probe policy robustness (LIBERO [19], REALM [28]); but on a limited task set. RoboLab's task generation and diagnostic analysis allows large-scale evaluation and performance analysis, complementary to existing benchmarks.

Real-to-sim Evaluation. Recent work have focused on leveraging 3D reconstruction to build photorealistic simulation scenes from real-world videos in order to achieve closer visual alignment between simulation and real world photorealism [16, 12, 10, 38]. These works typically use Gaussian splatting, 3D segmentation, and multi-view inpainting, often operated at a per-scene level, which entails costly optimization and makes it slow to scale beyond a small number of environments [10, 35, 27] (Appendix A-B). In contrast, our framework produces large-scale, photorealistic scenes and tasks within minutes rather than hours, while preserving sufficient geometric and visual fidelity for policy evaluation, thereby making real-

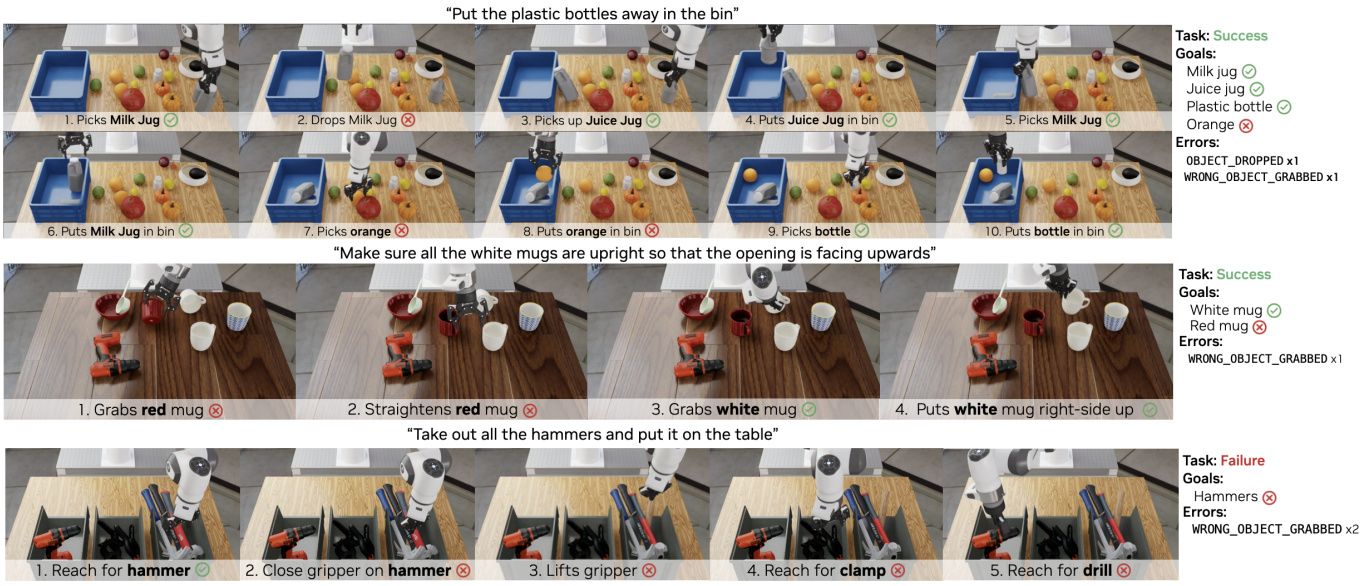


Fig. 3: **Task progression of a few tasks, illustrating errors encountered during policy rollout.** Top row: Although the task is successfully completed, errors were encountered during execution: 1) The robot drops the milk jug too early, missing the bin. 2) the robot grasps an orange (wrong object) and puts it in the bin. Mid row: An extraneous object was reoriented before the actual intended object. Final row: Intended objects were attempted unsuccessfully, and the policy tended to two wrong other objects.

to-sim benchmarking practical at the scale needed for modern generalist robot policies.

III. RoboLab

Evaluating real-world, generalist robotics policies in simulation remains a significant challenge. RoboLab aims to address this with a few design principles: 1) Enable easy and fast scene/task generation; 2) Tasks in RoboLab are policy- and robot- agnostic, enabling comparison of multiple policy and robot choices when it comes to solving real-world tasks; and 3) generate tasks that are diverse, representative of real-world tasks, and enable a multifaceted analysis of models to understand their generalization capabilities.

A. RoboLab Scene and Task Generation

RoboLab introduces a a user-friendly workflow that mirrors the process of preparing a real-world robot evaluation (Fig. 1): 1) First, create a **scene** by positioning and orienting objects in a workspace; 2) Then, define a **task** as language instructions for a goal state in the scene; 3) Lastly, instantiate an **environment** by selecting a robot, policy, and variations of scene features including camera, lighting, and backgrounds for a task. RoboLab enables evaluation of the same tasks across different robot embodiments. By deferring robot- and experiment-specific binding to runtime, we facilitate systematic evaluation of tasks across robot configurations and policy variants, without having embodiment-specific training.

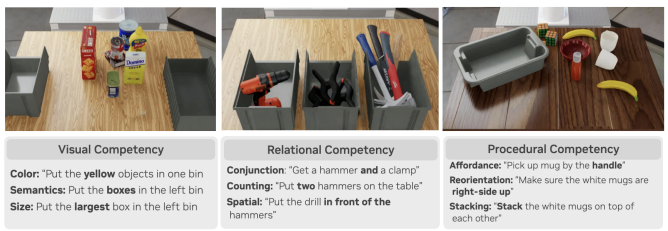


Fig. 4: Example of language instructions in RoboLab-120.

Formally, define a *scene* $S = \{(b_i, \mathbf{p}_i, \mathbf{q}_i)\}_{i=1}^{N}$, where b_i represents an object instance selected from the available catalog of objects $\mathcal{B}$ and $\mathbf{p}_i \in \mathbb{R}^3, \mathbf{q}_i \in SO(3)$ denote its position and orientation. Define a *task* $\mathcal{T} = \{S, l\}$, where l is the *language instruction* to complete in the scene. Define a policy $\pi : \mathcal{O} \to \mathcal{A}$ where the action space $\mathcal{A} \in \{\mathcal{A}^{\text{joint}}, \mathcal{A}^{\text{EE}}, \dots\}$ and observation space $\mathcal{O} = (\mathcal{O}^{\text{proprio}}, \mathcal{O}^{\text{rgb}}, \mathcal{O}^{\text{depth}} \dots)$ is policy dependent. An environment $E = (\mathcal{T}, \mathcal{R}, \mathcal{O}, \mathcal{A}, \xi)$ consists of a task, robot embodiment $\mathcal{R}$, policy parameters $(\mathcal{A}, \mathcal{O})$, and scene variations $\xi = (\xi^{\text{camera}}, \xi^{\text{light}}, \xi^{\text{background}}, \xi^{\text{pose}})$. More details on the specific objects, scenes and tasks in RoboLab can be found in Appendix A. To scale scene and task generation, we introduce an automated workflow, described below:

1) Scaling Scene Generation: We enable scaling scene generation through an automated pipeline that: 1) prompts an LLM to generate a structured scene plan for asset placement; 2) uses a geometric solver and physics simulation to check asset placement validity; and 3) refines the scene if it is not valid. First, the LLM is prompted with a theme (e.g., "messy counter") to generate a structured scene plan consisting of a subset of objects $B \subset \mathcal{B}$ and spatial predicates $\mathcal{P}$ governing the layout. The LLM is provided with the full catalog of objects $\mathcal{B}$ containing names and bounding box dimensions $\mathbf{d}_i \in \mathbb{R}^3$. Second, a spatial solver converts the relational predicates $\mathcal{P}$ into valid pose configurations $(\mathbf{p}, \mathbf{q})$). Objects are processed in dependency order, with support surfaces placed before objects on those surfaces (Algorithm 1). To check physical stability,

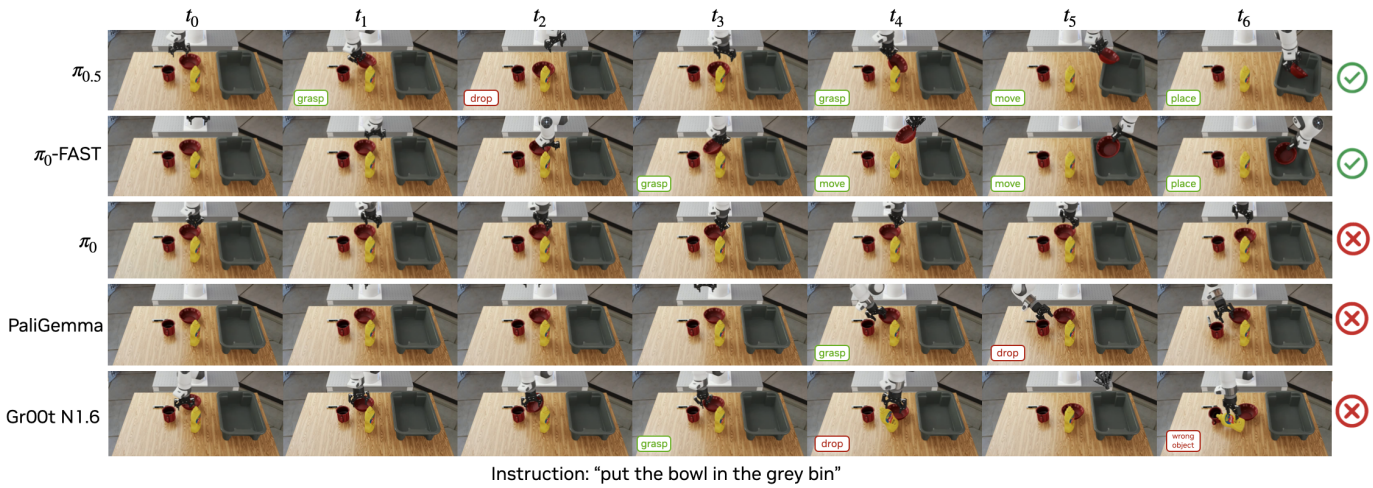


Fig. 5: Comparison of policy performance for bowl-in-bin manipulation. Rows represent distinct policies shown in chronological order (left to right). Successful execution involves grasping the central red bowl and depositing it into the gray bin on the right. Unsuccessful attempts are characterized by aimless arm trajectories and a lack of object interaction.

the scene is then forward simulated in Isaac Sim [22] for 300 steps under gravity. An object b_i is flagged as *unstable* if it's maximum Euclidean displacement is larger than a threshold (typically 0.02m). Third, If any object is unstable, we generate a text error describing the failure (e.g., "Object 'apple' fell off 'plate' with displacement 0.15m"). This feedback is provided to the LLM to refine the scene plan and repeat the process. Further details, including on the spatial and physical solvers, are provided in Appendix C.

2) Scaling Task Generation: We enable scaling task generation through an automated pipeline that: 1) generates task code from information including the scene and competency axes; 2) validates code syntax; 3) validates asset selections in the scene; and 4) refines the task if it is not valid. First, we prompt an LLM with detailed task information: 1) the scene object catalog B_S and metadata with dimensions; 2) task examples demonstrating the task structure; 3) the complete predicate library defining sub-task success and termination; 4) Competency-axes language templates with placeholders for objects, spatial verbs, and attributes; and 5) constraints including difficulty levels and physical feasibility requirements (e.g., containment size constraints, stacking stability). Second, tasks are check for syntax validity as code. Third, asset validation checks that all objects are not in the forbidden set and, for containment tasks (e.g., "place b_i inside b_j"), that inner objects fit inside containers with some clearance. Fourth, if validation fails, feedback is gathered into a fix prompt $\mathcal{Q}_{\text{fix}}$ that includes the original prompt $\mathcal{Q}$, the invalid output, and an error message $\mathcal{E}$ describing syntax errors or invalid asset references. The fix prompt is provided to the LLM to refine the task and repeat the process.

B. Benchmark Design

Inspired by Visual Question and Answering (VQA) benchmarks, our benchmark design enables enables evaluating specific competency axes:

Visual Competency: Assesses recognition of *color*, *semantics*, and *size*, capturing the policy's capability to link perceptual attributes with higher-level reasoning.
Procedural Competency: Evaluates the ability to perform tasks that involve action-oriented reasoning, including *affordances*, *reorientation*, or *stacking*.
Relational Competency: Tests understanding of language *conjunctions* (e.g., 'and', 'or'), *counting*, and *spatial* relationships, measuring how effectively the policy interprets multi-object instructions and scene structure.

Figure 4 shows examples of these questions accompanied by scene examples. Since any one task cannot be categorized as containing exactly evaluating one attribute, each task is labeled with one or more attributes belonging to one or more competencies.

In RoboLab, each task can be decomposed into a *sequential list of subtasks, each containing parallel events*. For example, the task *"Put the apple and orange on the plate, then put the banana in the bowl"* decomposes to subtasks `PickPlace(orange) & PickPlace(apple)` followed by subtask `PickPlace(banana)`, where each `PickPlace` contains parallel events (Grasp $\rightarrow$ Hover $\rightarrow$ Drop $\rightarrow$ Done) for each object.

Difficulty of tasks in our benchmarking system is rated using the following formula, based on the task length and require level of competency: $\text{DifficultyScore} = N_{\text{subtasks}} + \max(w_{\text{skill}})$ where w_{skill} is 0 for visual identification, 1 for spatial reasoning, 2 for procedural reasoning, and 3 for reorientation and dynamic tasks. We use this system based on how much reasoning and dexterity should be required in order for the task to be complete. Based on the difficulty score, tasks fall into one the following difficulty levels: *simple* (≤ 2), *moderate* (3–4), or *complex* (≥ 5). The difficulty levels allows the user to quickly analyze the results at a glance, although we provide a more granular analysis is possible by examining scene composition and task

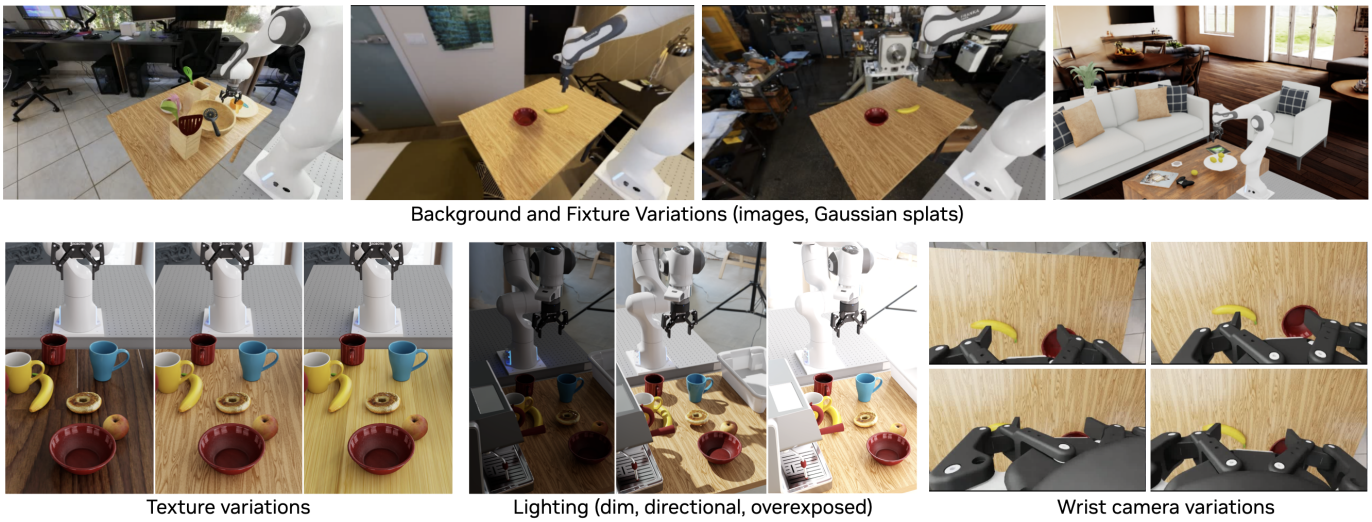


Fig. 6: Example scene variations, lighting variations, and camera pose variations in RoboLab.

horizon, discussed in Sec. IV-B.

C. Metrics for Evaluation

While task success rate remains a fundamental metric, prior work [14] has demonstrated that they fail to reveal nuanced aspects of policy behavior and failure modes. Unlike approaches relying on human judgment [12], we define a set of discrete and continuous metrics to characterize policy performance. All of the following measures are independent measures and together paint a complete picture of policy bias.

Normalized Scores. We compute a *normalized graded score* for each task based on the subtasks τ as follows: $Sc(\mathcal{T}) = \frac{1}{|\mathcal{T}|}\sum_{\tau\in\mathcal{T}} w_\tau Sc(\tau)$, where $Sc(\tau) = ||\sum_e w_e||$ is the subtask score given events e. By default, all events and subtask weights are equal. Subtask/event weights default to equal but are user-configurable per task, allowing consequential events (e.g., an initial grasp in a cluttered scene) to be weighted appropriately. Successful episodes will have a score of 1.

Language Variations. Tasks in RoboLab are paired with a set of language instructions that the evaluator can query from. Having a set of high-variance language instructions is crucial for task-generalist policies: Given language variants of the same underlying task, the policy should behave similarly. This provides insight into the reasoning failures.

Trajectory Metrics. Trajectory quality metrics capture characteristics of motion efficiency and optimality. We compute the following: *Spectral arc-length (SPARC)*, which evaluates motion smoothness [2] via the arc length of the normalized Fourier magnitude spectrum of the velocity profile. Given a speed profile $v(t)$ of the end effector over time interval $[0, T]$

$$\text{SPARC} = -\int_0^{\omega_c} \sqrt{\left(\frac{1}{\omega_c}\right)^2 + \left(\frac{d\hat{V}(\omega)}{d\omega}\right)^2}\, d\omega \qquad (1)$$

where $\hat{V}(\omega) = V(\omega)/V(0)$ represents the normalized Fourier magnitude spectrum. Smoother motions yield values closer to zero, while jerkier trajectories produce more negative values. We employ an adaptive cutoff frequency $\omega_c = \min(10\text{ Hz}, \omega_\alpha)$, where $\omega_\alpha = \max_{k\in\mathcal{K}} \omega_k$ and $\mathcal{K} = \{k \mid \hat{V}(\omega_k) \geq \alpha\}$ denotes the set of frequency bins exceeding threshold $\alpha = 0.05$. This adaptive strategy ensures that the smoothness evaluation focuses on relevant frequency components. Lastly, trajectory optimality is assessed through end effector *speed* $v(t)$, and *path length* $l = \sum_{k=0}^{N-1} \|p_{k+1} - p_k\|$, where p_k denotes the end-effector position at timestep k. Shorter path lengths indicate more direct trajectories and generally reflect superior motion quality.

D. Sensitivity Analysis

We present a Bayesian framework for evaluating policy robustness across diverse environmental conditions using Simulation-Based Inference (SBI). This analysis provides insight into which scene parameters are most strongly linked to success and failure outcomes by learning an approximate posterior distribution over them given evaluation data. Let $\theta = (\theta^{\text{cont}}, \theta^{\text{disc}})$ denote the environment parameters comprising of continuous variables (e.g., object distance, camera displacement) and/or discrete variables. After evaluating policy π under varied conditions, we generate episodes $\mathcal{D} = \{(\theta_i, x_i)\}_{i=1}^N$ with observed outcomes x_i (e.g., task success). The posterior distribution $p(\theta \mid x) \propto p(x \mid \theta)p(\theta)$ is approximated using Mixed Neural Posterior Estimation (MNPE), which trains a neural density estimator $q_\phi(\theta \mid x)$ to directly learn the mapping from observations to parameter distributions. The resulting posterior $q_\phi(\theta \mid x)$ characterizes which scene variables are most associated with a target observation x. Our approach provides systematic assessment of which variables most strongly influence performance outcomes. Further details are in Appendix B.

Task Adherence. While task success is determined by the target world state, the policy may exhibit undesirable behaviors during rollout that do not affect the final outcome.

TABLE I: **Overall performance of SOTA policies on RoboLab.** While recent foundation models exhibit emerging capabilities across diverse task dimensions, overall performance remain limited. See Table IV and Table V for more granular breakdown of performance trajectory-quality results.

Model	Overall Metrics			Difficulty (succ% / score)			Procedural (succ% / score)			Relational (succ% / score)			Visual (succ% / score)		
	Succ% / Score (↑)	SPARC (↑)	Speed (↑)	simple	moderate	complex	affordance	reorientation	stacking	conjunction	counting	spatial	color	semantics	size
$\pi_{0.5}$ [9]	**28.0 / 0.43**	−8.34 ±6.7	**5.4** ±1.6	**29.7 / 0.39**	**31.5 / 0.51**	**13.5 / 0.44**	**10.0 / 0.27**	**28.3 / 0.48**	**35.0 / 0.57**	**43.8 / 0.55**	**65.7 / 0.75**	**21.0 / 0.36**	**23.8 / 0.42**	**23.2 / 0.39**	**30.0 / 0.36**
π_0-FAST [26]	15.5 / 0.27	−9.63 ±5.4	4.6 ±1.8	20.2 / 0.27	13.3 / 0.29	2.9 / 0.22	2.5 / 0.13	3.3 / 0.10	15.0 / 0.28	27.5 / 0.38	44.3 / 0.59	15.9 / 0.30	6.2 / 0.17	10.5 / 0.22	10.0 / 0.18
GR00T N1.6 [23]	7.2 / 0.17	**−6.87** ±4.6	4.3 ±1.4	8.8 / 0.14	7.9 / 0.25	0.0 / 0.12	0.8 / 0.11	0.0 / 0.03	8.3 / 0.18	10.0 / 0.14	18.6 / 0.34	7.2 / 0.22	3.8 / 0.16	6.8 / 0.17	0.0 / 0.03
π_0 [5]	5.0 / 0.12	−9.49 ±4.1	4.2 ±1.3	7.2 / 0.10	3.6 / 0.18	0.0 / 0.09	0.0 / 0.07	1.7 / 0.06	0.0 / 0.05	12.5 / 0.21	11.4 / 0.28	3.1 / 0.16	1.2 / 0.08	2.3 / 0.09	1.7 / 0.03
PaliGemma [4]	3.4 / 0.10	−16.52 ±10.2	1.9 ±1.6	3.4 / 0.06	4.9 / 0.16	0.0 / 0.09	0.0 / 0.01	1.7 / 0.07	0.0 / 0.06	3.8 / 0.06	22.9 / 0.36	1.7 / 0.12	0.8 / 0.09	3.5 / 0.09	0.0 / 0.05

For example, Fig. 3 demonstrates a successful episode in which the policy incorrectly grasped an extraneous object. Such errors highlight potential biases in the policy not captured by other metrics. Thus, capturing discrete events such as grasping wrong objects, executing redundant actions and unintended collisions, highlights reasoning and task adherence failures. Our benchmark automatically records instances of events; including wrong object grasped, object dropped, and gripper collisions.

IV. Experiments

We evaluate several off-the-shelf generalist robotics models, performing controlled ablations, and environmental perturbations to observe where failures concentrate. The experiments are designed to address the following questions: **Q1:** How well does today's SOTA models generalize, given varying language instructions, scene complexity, and environment pertubations? **Q2:** When and why does a policy fail? **Q3:** Can a simulated benchmark be used to evaluate real-world models?

A. Experiment Setup

We evaluate SOTA models on RoboLab-120, a benchmark containing 120 tasks of varying difficulty levels (65 simple, 38 moderate, 18 complex). We evaluate policies finetuned on the DROID robot [13], which is an open-sourced robot used for VLAs [10, 1]. DROID has a 7-DOF Franka Panda robot arm with a Robotiq-2F-85 gripper, an externally mounted ZED 2i camera with f=2.1mm, and a ZED mini as the wrist camera. All policies were fine-tuned on the DROID dataset [13]: $\pi_{0.5}$ [9], π_0-FAST [26], π_0 [5], PaliGemma [4], GR00T N1.6 [23].

The action space is 7-DOF Franka joint positions and a 1-DOF binary gripper command. The environments are composed of a default office-like background and natural lighting to mimic typical setups in the DROID dataset [13], with wrist and external camera poses designed to strongly match the real-world DROID robot. Each task was repeated with N=10 episodes per task. Notes on statistical analysis of N is discussed in Appendix A-A.

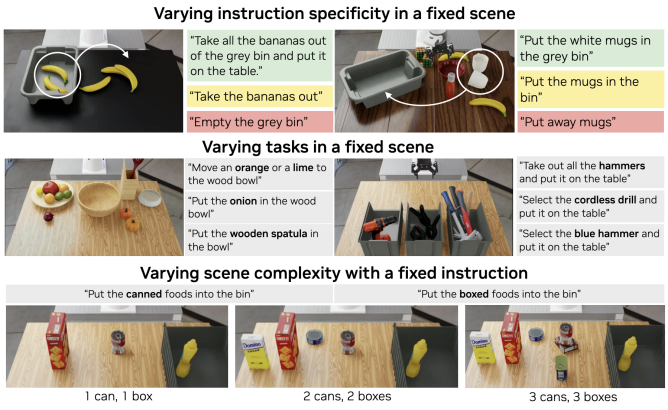


Fig. 7: Examples of language ablation experiments. Top: Same scene and goal, but the instruction wording ranges from precise to increasingly vague. Middle: Same scene, but the instruction specifies different tasks to perform. Bottom: Same instruction, but the scene becomes progressively more complex.

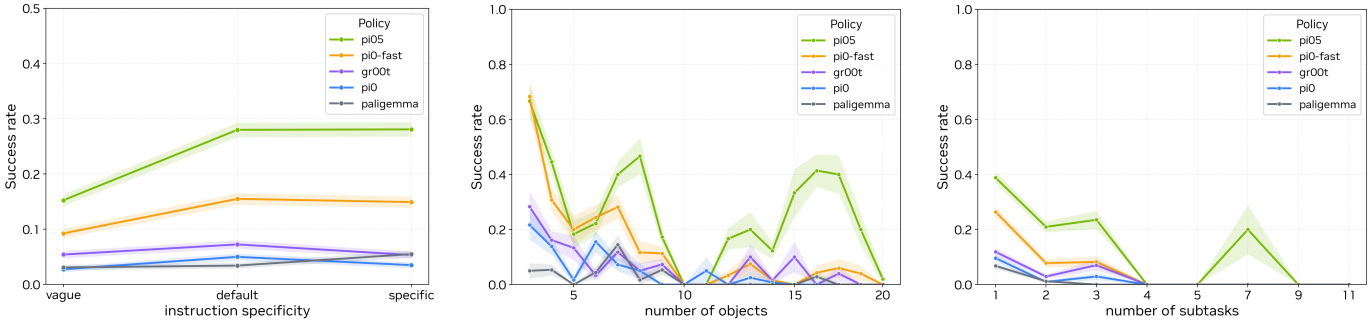
B. Task Results

Table I shows the overall results on RoboLab-120, highlighting success rate and score. Note that success rates indicates number of episodes that were completed, whereas the score indicates whether if the policy was able to make progress on any task. Overall success were low, which matches prior observations on out-of-domain generalization for generalist policies [36]. More interestingly, the score/success gap reveals capabilities that raw success rates obscure. $\pi_{0.5}$ achieves only 13.5% success on complex tasks yet attains a score of 0.44, indicating that nearly half of the partial-credit milestones are reached even when full task completion is rare. This indicates that contemporary policies often partially understand the task but fail in the final stages of execution.

Competency axes also highlighted asymmetric generalization in *relational* reasoning tasks. Overall, policies handled conjunctions and counting better than spatial relations. In *visual* grounding, performance remained low across attribute types, indicating brittle language-to-object binding beyond a narrow set of familiar object descriptions. *Procedural* understanding proved most challenging.

To further probe generalization robustness of the policies, we performed analysis on variations on instructions, scene complexity, and task difficulty. An example of these experiments is shown in Fig. 7.

Performance relative to instruction specificity. Fig. 8a reports overall RoboLab-120 success rates under three levels of instruction specificity (vague, default, specific) for the same underlying tasks and scenes. We observe degradation as instructions become more abstract (e.g., $\pi_{0.5}$ drops from 28.0% on default to 15.3% on vague) This sensitivity reveals that current models lack robust reasoning against the task goal. A truly generalist policy should be invariant to instruction phrasing. A policy that has understood the task should perform comparably whether the instruction is vague or specific.



(a) SR relative to instruction specificity (b) SR relative to scene complexity (c) SR relative to task horizon

Fig. 8: **Effect of language, scene complexity, and task horizon on policy performance (success rate).** Performance using success rate across complexity axes. For a robust policy, performance should relatively comparable as complexity increases. (a) Performance degrades as instructions become more abstract, indicating brittle task-level reasoning. (b) Performance degrades sharply as the number of objects in the scene increases, exposing systematic geometric biases in current models. (c) Performance degrades as the task horizon increases, indicating limited multi-step reasoning across all evaluated policies. Fig. 12 shows the same analysis with normalized scores for a more granular analysis.

Performance relative to scene complexity. Fig. 8b isolates the effect of scene complexity by increasing the number of objects visually present on the table, representing visual clutter. Success rates degrade as the number of objects in the scene increases for most policies; however, for $\pi_{0.5}$, some tasks were able to be achieved even with high visual clutter. These results indicates that current policies can perform some visual grounding in cluttered scenes; however, for some policies, additional distractors overwhelm the policy's ability to identify and manipulate the correct target.

Performance relative to task horizon. Fig. 8c shows that performance degrades as the task horizon increases. Performance degrades as the task horizon increases. This indicates that current models lack the multi-step reasoning and compositional planning required for longer-horizon tasks. A performance increase is observed at subtask=7 for $\pi_{0.5}$ on `CubesAndBlocksInBinTask` is further discussed in Appendix A-D.

Together, these results show how RoboLab isolates where generalization fails, addressing **Q1** supporting analysis that can inform data collection and model improvement priorities.

C. Sensitivity Analysis

We perform a set of variations given two basic tasks and observe the outcome, as illustrated in Fig. 6. These are: 1) Wrist and external camera poses; 2) object poses; 3) Visual features, including background and table textures; and 4) Lighting, including saturation and hue. Table II illustrates the results for all experiments.

Visual and Lighting variations. We vary the lighting conditions via color temperature shifts, lighting exposure and strong directional light that generates shadows as the robot is moving. **Lighting:** Models were robust to changes in lighting conditions, with 90–100% success across shadow variations, color temperature shifts, and 500× intensity changes. **Visual appearance:** Variations over 10 background textures and 4 table textures had minimal impact (<5% degradation), suggesting generalization to scene appearance changes.

Camera variation sensitivity analysis. We infer posteriors over camera displacement conditioned on task success (Fig. 9). Camera poses were randomized in both orientation and position for 10 episodes each. Displacement is calculated with respect to the nominal position of the cameras. Across all policies, the wrist-camera posterior is sharply concentrated near zero, indicating that successful execution often required the wrist camera to remain close to its nominal pose, while performance is more tolerant to external camera position changes. This indicates performance is critically dependent on wrist camera than external camera.

Object pose variation sensitivity analysis. We randomize initial object poses via a uniform distribution of 10cm, 20cm, and 30cm within its nominal placement (in front of the robot). We then infer posteriors over initial object poses conditioned on task success (see Fig. 9), relative to the robot pose. We observe a strong peak over 0.5m from the robot's origin, suggesting that objects placed at this distance has the highest probability of success, likely due to reachability.

Together, these results show how RoboLab's analysis framework can be used to systematically identify factors contributed to policy failure, addressing **Q2**.

D. Real-World Verification

To assess whether RoboLab can be used as a proxy for real-world evaluation, we compare performance results from RoboLab against RoboArena [1], an open-source real-world benchmarking system. Fig. 10 plots per-policy RoboLab-120 success rate against RoboArena Elo scores. We observe that the ranking between policies is strongly preserved (Spearman $\rho = 1.00$). Because RoboArena reports Elo while RoboLab reports success rates, Spearman rank correlation is more useful as a proxy, since it measures whether the two benchmarks induce the same ordering over policies. This indicates that RoboLab achieves benchmark-level correlation with real-world

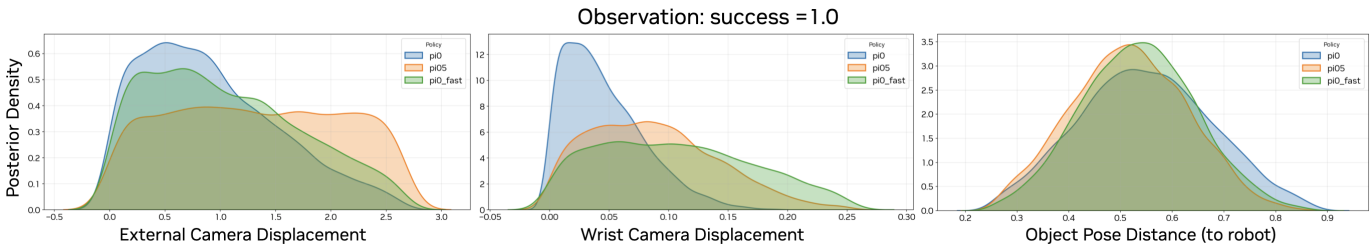


Fig. 9: Results of the sensitivity analysis using MNPE. Policies were highly sensitive to wrist-camera displacement from the nominal pose, indicating strong dependence on wrist-mounted camera calibration. Success also peaked for objects placed at approximately 0.5m from the robot, likely due to robot reachability.

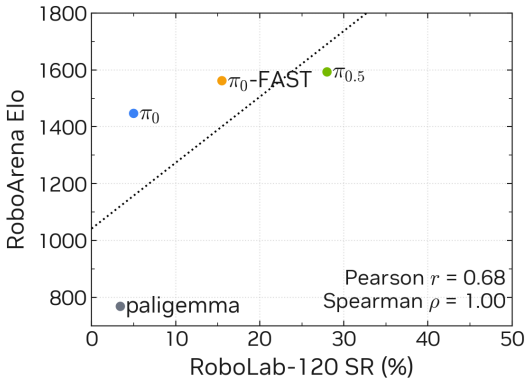


Fig. 10: Correlation between RoboLab-120 success rate and RoboArena Elo score across the four policies with both measurements available ($\pi_{0.5}$, π_0-FAST, π_0, PaliGemma). Rankings are preserved (Spearman $\rho = 1.00$) and the scores are positively correlated (Pearson $r = 0.68$), suggesting RoboLab-120 performance is a useful proxy for real-world policy quality.

TABLE II: Robustness to controlled environmental variations over two simple tasks (`BananaInBowl`, `BananaAndCubeInBowl`). PaliGemma is excluded as it fails to achieve meaningful results.

	$\pi_{0.5}$		π_0**-FAST**		π_0	
Variation	Succ.%	Time (s)	Succ.%	Time (s)	Succ.%	Time (s)
Lighting						
Color	96.7	14.5 ±7.9	93.3	17.9 ±10.7	6.7	31.1 ±4.3
Shadows	100.0	16.0 ±6.0	90.0	12.4 ±3.5	0.0	-
Dim	90.0	9.1 ±2.1	70.0	13.1 ±2.8	70.0	35.5 ±9.7
Overexposed	100.0	13.9 ±4.4	100.0	9.6 ±1.7	0.0	-
Visual Variations						
Background	85.0	14.4 ±8.7	70.0	21.3 ±10.7	25.0	31.6 ±11.8
Table texture	87.5	19.0 ±13.8	60.0	19.0 ±12.9	22.5	28.1 ±6.9
Object Pose						
10cm	95.0	16.2 ±9.2	55.0	26.5 ±13.8	22.5	34.7 ±13.3
20cm	95.0	19.7 ±10.2	40.0	21.8 ±9.5	20.0	37.4 ±11.2
30cm	62.5	18.9 ±8.7	35.0	24.3 ±11.9	17.5	24.3 ±11.9
Camera Pose						
external	85.0	17.4 ±11.3	45.0	27.7 ±10.6	50.0	27.4 ±16.0
wrist	60.0	21.9 ±13.4	25.0	20.1 ±9.9	10.0	35.3 ±10.4

performance, addressing **Q3**. We leave deeper investigation of task-level and motion-level correlation to future work.

V. Limitations

While RoboLab provides a flexible and scalable framework for evaluating language-conditioned manipulation, it currently focuses on rigid-body tabletop scenes and does not fully capture the challenges of deformable object manipulation (e.g., cloth, cables, bags). Moreover, many contact-rich skills that require precise force control, compliant interaction, or complex frictional dynamics are underrepresented and dependent on the physics simulation fidelity, limiting Robolab's coverage of fine-grained, low-level control tasks. Our subtask evaluation system breaks down for open-ended and ambiguous long-horizon tasks (e.g., "clean up all the laundry") and human judgment becomes necessary. However, given that current policies achieve fairly low scores, this is not yet a practical bottleneck, and our approach remains valid. Finally, although evaluation in high-fidelity simulation is a strong proxy for real-world performance, a residual visual distribution shift remains. This gap needs to be characterized further both by analyzing the behavior and robustness of the visual perception stack and through extensive validation on real-world deployments.

VI. Conclusion

Recent benchmarking efforts have made significant strides in scalable robot evaluation, but they primarily assess robustness to perturbations of training environments rather than true task generalization to novel scenarios. RoboLab addresses this gap by evaluating real world policies in a high-fidelity simulation, structured evaluation vectors that decompose policy competence into visual, procedural, and relational dimensions, and a set of sensitivity analysis set of novel analysis that provides insight into policy behavior for robotics. Our benchmarking framework enables the community to critically answer the question of *generalization* and *performance*. At the same time, the framework is designed to be pragmatically usable: new tasks can be authored in minutes by arranging objects on a tabletop and attaching language instructions, and a generative scene–task–environment workflow that supports continuous benchmark evolution.

References

[1] Pranav Atreya, Karl Pertsch, Tony Lee, Moo Jin Kim, Arhan Jain, Artur Kuramshin, Clemens Eppner, Cyrus Neary, Edward Hu, Fabio Ramos, et al. Roboarena: Distributed real-world evaluation of generalist robot policies. In *Proceedings of the Conference on Robot Learning (CoRL 2025)*, 2025.

[2] Sivakumar Balasubramanian, Alejandro Melendez-Calderon, and Etienne Burdet. A robust and sensitive metric for quantifying movement smoothness. *IEEE Transactions on Biomedical Engineering*, 59(8): 2126–2136, 2012. doi: 10.1109/TBME.2011.2179545.

[3] Prithviraj Banerjee et al. HOT3D: Hand and object tracking in 3D from egocentric multi-view videos. *CVPR*, 2025.

[4] Lucas Beyer, Andreas Steiner, André Susano Pinto, Alexander Kolesnikov, Xiao Wang, Daniel Salz, Maxim Neumann, Ibrahim Alabdulmohsin, Michael Tschannen, Emanuele Bugliarello, Thomas Unterthiner, Daniel Keysers, Skanda Koppula, Fangyu Liu, Adam Grycner, Alexey Gritsenko, Neil Houlsby, Manoj Kumar, Keran Rong, Julian Eisenschlos, Rishabh Kabra, Matthias Bauer, Matko Bošnjak, Xi Chen, Matthias Minderer, Paul Voigtlaender, Ioana Bica, Ivana Balazevic, Joan Puigcerver, Pinelopi Papalampidi, Olivier Henaff, Xi Xiong, Radu Soricut, Jeremiah Harmsen, and Xiaohua Zhai. Paligemma: A versatile 3b vlm for transfer, 2024. URL https://arxiv.org/abs/2407.07726.

[5] Kevin Black, Noah Brown, Danny Driess, Adnan Esmail, Michael Equi, Chelsea Finn, Niccolo Fusai, Lachy Groom, Karol Hausman, Brian Ichter, Szymon Jakubczak, Tim Jones, Liyiming Ke, Sergey Levine, Adrian Li-Bell, Mohith Mothukuri, Suraj Nair, Karl Pertsch, Lucy Xiaoyang Shi, James Tanner, Quan Vuong, Anna Walling, Haohuan Wang, and Ury Zhilinsky. π_0: A vision-language-action flow model for general robot control, 2026. URL https://arxiv.org/abs/2410.24164.

[6] Rishit Dagli, Donglai Xiang, Vismay Modi, Charles Loop, Clement Fuji Tsang, Anka He Chen, Anita Hu, Gavriel State, David I.W. Levin, and Maria Shugrina. Vomp: Predicting volumetric mechanical property fields. *arXiv preprint*, 2025.

[7] Jiayuan Gu, Fanbo Xiang, Xuanlin Li, Zhan Ling, Xiqiang Liu, Tongzhou Mu, Yihe Tang, Stone Tao, Xinyue Wei, Yunchao Yao, et al. Maniskill2: A unified benchmark for generalizable manipulation skills. *arXiv preprint arXiv:2302.04659*, 2023.

[8] Andrew Guo, Bowen Wen, Jianhe Yuan, Jonathan Tremblay, Stephen Tyree, Jeffrey Smith, and Stan Birchfield. HANDAL: A dataset of real-world manipulable object categories with pose annotations, affordances, and reconstructions. In *IROS*, 2023.

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

[10] 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, 2025. URL https://arxiv.org/abs/2512.16881.

[11] Stephen James, Zicong Ma, David R. Arrojo, and Andrew J. Davison. RLBench: The Robot Learning Benchmark & Learning Environment. *RAL*, 2020.

[12] Yash Jangir, Yidi Zhang, 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, 2025. URL https://arxiv.org/abs/2510.23571.

[13] Alexander Khazatsky, Karl Pertsch, Suraj Nair, Ashwin Balakrishna, Sudeep Dasari, Siddharth Karamcheti, Soroush Nasiriany, Mohan Kumar Srirama, Lawrence Yunliang Chen, Kirsty Ellis, et al. DROID: A large-scale in-the-wild robot manipulation dataset. In *Robotics: Science and Systems (RSS)*, 2024.

[14] Hadas Kress-Gazit, Kunimatsu Hashimoto, Naveen Kuppuswamy, Paarth Shah, Phoebe Horgan, Gordon Richardson, Siyuan Feng, and Benjamin Burchfiel. Robot learning as an empirical science: Best practices for policy evaluation, 2024. URL https://arxiv.org/abs/2409.09491.

[15] Chengshu Li, Ruohan Zhang, Josiah Wong, Cem Gokmen, Sanjana Srivastava, Roberto Martín-Martín, Chen Wang, Gabrael Levine, Wensi Ai, Benjamin Martinez, et al. Behavior-1k: A human-centered, embodied ai benchmark with 1,000 everyday activities and realistic simulation. *arXiv preprint arXiv:2403.09227*, 2024.

[16] Xuanlin Li et al. Evaluating real-world robot manipulation policies in simulation. *arXiv preprint arXiv:2405.05941*, 2024.

[17] Yunzhi Lin, Jonathan Tremblay, Stephen Tyree, Patricio A. Vela, and Stan Birchfield. Multi-view fusion for multi-level robotic scene understanding. In *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 6817–6824, 2021. doi: 10.1109/IROS51168.2021.9635994.

[18] Zhiqiu Lin, Deepak Pathak, Baiqi Li, Jiayao Li, Xide Xia, Graham Neubig, Pengchuan Zhang, and Deva Ramanan. Evaluating text-to-visual generation with image-to-text generation, 2024. URL https://arxiv.org/abs/2404.01291.

[19] Bo Liu, Yifeng Zhu, Chongkai Gao, Yihao Feng, Qiang Liu, Yuke Zhu, and Peter Stone. Libero: Benchmarking knowledge transfer for lifelong robot learning. *Advances in Neural Information Processing Systems*, 36:44776–44791, 2023.

[20] Oier Mees, Lukas Hermann, Erick Rosete-Beas, and Wolfram Burgard. CALVIN: A Benchmark for Language-Conditioned Policy Learning for Long-Horizon Robot Manipulation Tasks. *IEEE Robotics and Automation*

Letters, 7(3):7327–7334, 2022.

[21] Nicolas Moenne-Loccoz, Ashkan Mirzaei, Or Perel, Riccardo de Lutio, Janick Martinez Esturo, Gavriel State, Sanja Fidler, Nicholas Sharp, and Zan Gojcic. 3d gaussian ray tracing: Fast tracing of particle scenes. *ACM Transactions on Graphics and SIGGRAPH Asia*, 2024.

[22] NVIDIA. Isaac Sim. URL https://github.com/isaac-sim/IsaacSim.

[23] NVIDIA, Johan Bjorck, Nikita Cherniadev Fernando Castañeda, Xingye Da, Runyu Ding, Linxi "Jim" Fan, Yu Fang, Dieter Fox, Fengyuan Hu, Spencer Huang, Joel Jang, Zhenyu Jiang, Jan Kautz, Kaushil Kundalia, Lawrence Lao, Zhiqi Li, Zongyu Lin, Kevin Lin, Guilin Liu, Edith Llontop, Loic Magne, Ajay Mandlekar, Avnish Narayan, Soroush Nasiriany, Scott Reed, You Liang Tan, Guanzhi Wang, Zu Wang, Jing Wang, Qi Wang, Jiannan Xiang, Yuqi Xie, Yinzhen Xu, Zhenjia Xu, Seonghyeon Ye, Zhiding Yu, Ao Zhang, Hao Zhang, Yizhou Zhao, Ruijie Zheng, and Yuke Zhu. GR00T N1: An open foundation model for generalist humanoid robots. In *ArXiv Preprint*, March 2025.

[24] OpenAI. Gpt-4 technical report, 2024. URL https://arxiv.org/abs/2303.08774.

[25] OpenAI. Openai o1 system card, 2024. URL https://arxiv.org/abs/2412.16720.

[26] Karl Pertsch, Kyle Stachowicz, Brian Ichter, Danny Driess, Suraj Nair, Quan Vuong, Oier Mees, Chelsea Finn, and Sergey Levine. Fast: Efficient action tokenization for vision-language-action models, 2025. URL https://arxiv.org/abs/2501.09747.

[27] Mohammad Nomaan Qureshi, Sparsh Garg, Francisco Yandun, David Held, George Kantor, and Abhishesh Silwal. Splatsim: Zero-shot sim2real transfer of rgb manipulation policies using gaussian splatting, 2024. URL https://arxiv.org/abs/2409.10161.

[28] Miroslav Sedlacek et al. Realm: A real-world benchmark for evaluating robot learning methods. *arXiv preprint arXiv:2512.19562*, 2024. URL https://arxiv.org/abs/2512.19562.

[29] Isaac Lab Team. Isaac Lab: A GPU-accelerated simulation framework for multi-modal robot learning. *arXiv preprint arXiv:2511.04831*, 2025. URL https://arxiv.org/abs/2511.04831.

[30] TRI LBM Team. A careful examination of large behavior models for multitask dexterous manipulation, 2025. URL https://arxiv.org/abs/2507.05331.

[31] Qi Wu, Janick Martinez Esturo, Ashkan Mirzaei, Nicolas Moenne-Loccoz, and Zan Gojcic. 3dgut: Enabling distorted cameras and secondary rays in gaussian splatting. *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2025.

[32] Yu Xiang, Tanner Schmidt, Venkatraman Narayanan, and Dieter Fox. Posecnn: A convolutional neural network for 6d object pose estimation in cluttered scenes. 2018.

[33] Yandan Yang, Baoxiong Jia, Shujie Zhang, and Siyuan Huang. Sceneweaver: All-in-one 3d scene synthesis with an extensible and self-reflective agent, 2025. URL https://arxiv.org/abs/2509.20414.

[34] Tianhe Yu, Deirdre Quillen, Zhanpeng He, Ryan Julian, Avnish Narayan, Hayden Shively, Adithya Bellathur, Karol Hausman, Chelsea Finn, and Sergey Levine. Meta-world: A benchmark and evaluation for multi-task and meta reinforcement learning, 2021. URL https://arxiv.org/abs/1910.10897.

[35] 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. *arXiv preprint arXiv:2511.04665*, 2025.

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

[37] Yuke Zhu, Josiah Wong, Ajay Mandlekar, Roberto Martín-Martín, Abhishek Joshi, Kevin Lin, Soroush Nasiriany, and Yifeng Zhu. robosuite: A modular simulation framework and benchmark for robot learning. In *arXiv preprint arXiv:2009.12293*, 2020.

[38] Alex Zook, Fan-Yun Sun, Josef Spjut, Valts Blukis, Stan Birchfield, and Jonathan Tremblay. Grs: Generating robotic simulation tasks from real-world images. In *Proceedings of the Computer Vision and Pattern Recognition Conference*, pages 594–603, 2025.