MolmoSpaces-CommonSense: Evaluating Commonsense in Robotic Manipulation Policies

Arjun Guru[1,2] Max Argus[2] Jae Sung Park[2] Jiafei Duan[1] Ali Farhadi[1] Ranjay Krishna[1]
[1]University of Washington [2]Allen Institute for AI
{aguru, duanj1, ranjay}@cs.washington.edu, ali@cs.uw.edu, {maxa, jamesp}@allenai.org

***Abstract*—Real-world robotic manipulation requires reasoning about cluttered scenes, partially observable or hidden objects, and semantically appropriate actions. However, existing simulation benchmarks remain dominated by comparatively clean environments with limited clutter and narrowly scoped task structure, leaving many forms of commonsense reasoning untested. We address this by introducing a taxonomy of reasoning processes for robotic manipulation grounded in developmental cognitive science. We operationalize this taxonomy through a new benchmark of robot tasks, *MolmoSpaces-CommonSense*, that requires reasoning about occlusion, containment, support relations, semantically correct grasp points, spatial affordances, and hidden objects. These evaluations require policies not only to recognize relevant scene structure but also to execute correct actions in them. We evaluate several zero-shot generalist manipulation policies. Across model types and architectures, we observe consistent failures in commonsense reasoning: success rates in cluttered scenes drop by over 50% as distractors increase, policies act on incorrect objects in cluttered settings 30% of the time, and affordance-sensitive grasping remains unreliable even with specific language prompts, with correct part selection only about 50% of the time. Further, VLM-based success estimation struggles to reliably evaluate rollouts in these settings, suggesting that the challenge extends beyond control to the broader perception-reasoning-action stack. Our results suggest that existing benchmarks may substantially overestimate zero-shot real-world readiness. Code and benchmark will be released at github.com/allenai/molmospaces.**

***Index Terms*—Robot Manipulation Benchmark, Commonsense Reasoning, Manipulation Policies**

I. Introduction

Manipulation in the real world is a reasoning problem before it is a control problem. A robot must decide which object an instruction refers to in clutter, infer what remains true when objects become partially hidden, determine what actions the scene affords, and choose behaviors that are physically and semantically appropriate for the goal. Cognitive science has long treated these capabilities not as a single faculty but as interacting forms of physical reasoning, affordance perception, and object permanence [2, 15, 36]. Yet the benchmarks that currently anchor progress in robot learning [3, 35, 21, 45, 44] are still built primarily around executing predefined task suites in comparatively controlled scenes with a small set of task-relevant objects, rather than systematically requiring reasoning under clutter, occlusion, or hidden state.

This gap is ubiquitous in homes, labs, and workspaces, which are cluttered and where targets are often occluded, covered, contained, or displaced out of view. Yet current robot policies are largely neither trained nor evaluated in environments that make these capabilities necessary [24]: a policy that succeeds only when the target is cleanly visible and the correct action is obvious is not one that can be deployed in the real world.

In this work, we introduce a taxonomy of commonsense reasoning processes for robotic manipulation, grounded in decades of cognitive science. The taxonomy (see Fig. 1) organizes a practically important subset of embodied commonsense into three categories: *physical reasoning* (object presence and relations under occlusion, covering, and containment), *affordances* (grasping affordances at the object-part level and spatial affordances at the surface level), and *object permanence* (acting on objects seen but no longer visible). Our goal is not to claim completeness but to provide a concrete framework for asking which reasoning processes current manipulation policies actually operationalize.

We operationalize this taxonomy by extending MolmoSpaces [24] with environments that force these reasoning processes to matter at action time. MolmoSpaces provides a scalable ecosystem of diverse indoor scenes, richly annotated objects, and infrastructure with strong sim-to-real alignment. On top of it, we construct tasks and conditions that demand reasoning at action time, such as dense distractor clutter, targets hidden beneath covers or inside containers, and prompts requiring semantically correct grasp points. Because these evaluations are closed-loop, success requires more than recognizing the right relation in an image; the policy must use that relation to produce the right action sequence.

We use MolmoSpaces-CommonSense to evaluate representative zero-shot generalist policies spanning both VLA and WAM paradigms, including $\pi_{0.5}$ [21], MolmoAct2 [12], and DreamZero [47]. The pattern is consistent across architectures. In cluttered physical-reasoning tasks, success drops by more than 50% as nearby distractors increase, and the dominant failure mode is not generic instability but selecting or moving toward the wrong object. In affordance-sensitive grasping tasks, overall success remains low, around 25%, and increasing prompt specificity barely improves whether the object is grasped correctly. Together, these results suggest current policies can execute atomic motions once the scene is easy, but do not yet reliably reason through the conditions that make manipulation hard. We further find that VLM judges struggle to reliably classify success and failure from rollout videos in

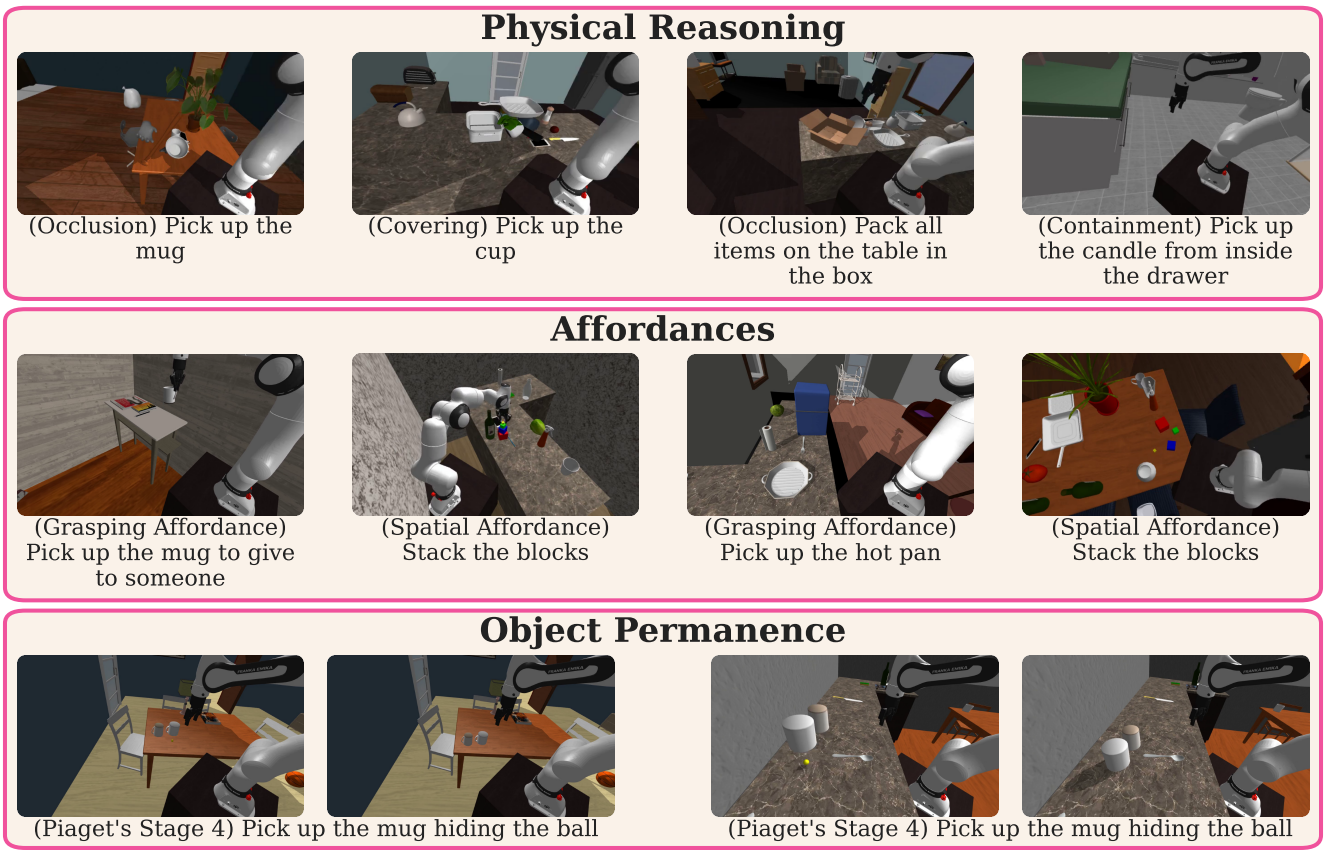

Fig. 1: Taxonomy of embodied commonsense reasoning capabilities, showing the diversity of tasks we test on. We organize tasks along three categories grounded in developmental cognitive science, and instantiate tasks within MolmoSpaces environments.

these scenes, suggesting that the difficulty extends across the broader perception-reasoning-action stack.

More broadly, this study is about what we choose to measure. If benchmarks do not test hidden-object reasoning, affordance-sensitive action selection, or behavior under partial observability, then high success rates can tell an overly optimistic story about readiness for deployment. By grounding benchmark design in cognitive science and instantiating that design in scalable embodied environments, we hope to make commonsense reasoning a first-class target for robot policy evaluation and improvement.

II. Related Work

Evaluating the reasoning capabilities of generalist robot policies draws on three lines of work: commonsense reasoning evaluation in foundation models, embodied reasoning benchmarks for robot policies, and simulation environments that make such evaluations reproducible and scalable.

Commonsense Reasoning Evaluation of Foundation Models. Commonsense reasoning has been studied extensively in language and vision-language models [41, 49, 50, 40, 34, 19, 29, 20], often via taxonomy-based or adversarially-filtered benchmarks, which inspire our taxonomy-driven evaluation of robot policies. Concerning physical and embodied reasoning, IntPhys [39] focuses on evaluating intuitive physics understanding, and Embodied Reasoning Question Answering [45] evaluates reasoning about actions, trajectories, 3D relationships, affordances, and more. Though a significant step, the goal of assessing embodied agents' reasoning capabilities ultimately requires closing the loop between reasoning and action and evaluating models' capabilities while they interact with the world. Our work aims to bridge this gap and test model capabilities in embodied settings beyond visual question-answering.

Embodied Reasoning Evaluations. As generalist manipulation models improve at atomic actions, a number of works have begun evaluating their higher-level reasoning across different subsets of embodied reasoning tasks. Works such as Colosseum [37] primarily study perturbations to tasks in simulation along multiple axes and measure policy robustness. STAR-Gen [14] develops a taxonomy on visual, semantic, and behavioral generalization and tests policies in the real world. VLABench [51] tests a categorization of VLA capabilities spanning visual, language, and physical understanding. Additional targeted benchmarks for specific components of embodied reasoning, such as memory [7, 11], affordance [48, 5, 18], and spatial reasoning [4, 43] have been developed. However, existing taxonomies omit notable dimensions of human commonsense reasoning capabilities, whereas capability-specific benchmarks only evaluate certain dimensions in isolation.

Our work aims to build upon these efforts and provide a taxonomy grounded in cognitive science literature on human embodied commonsense reasoning capabilities, along with a set of tasks in simulation that allow for reproducible and scalable evaluations of these capabilities.

Simulation benchmarks for robot policies. Simulation-based benchmarks have gained adoption as a scalable, reproducible alternative to real-world evaluation for robot manipulation [22, 28, 30]. More recent benchmarks have expanded to include language grounding and generalization capabilities [28, 30, 13, 52, 32, 33], robustness to perturbations [37], a more extensive set of short- and long-horizon tasks within the household [26], and alignment with the real world [27]. A number of these benchmarks require training in simulation [28, 30, 26, 32], limiting their practicality in evaluating zero-shot policies. In contrast, MolmoSpaces [24] supports zero-shot evaluations of generalist robot policies in over 20K indoor scenes and over 22K interactable articulated and rigid objects [25, 8, 46]. Our work builds upon the MolmoSpaces ecosystem while extending the list of tasks beyond solely atomic actions to reflect our taxonomy on commonsense reasoning.

III. Methods

Guided by our cognitive science-grounded task taxonomy, we design MolmoSpaces-CommonSense, a benchmark testing embodied commonsense reasoning and use it to evaluate generalist manipulation policies. Fig. 2 illustrates our approach.

A. Commonsense Taxonomy

Inspired by seminal studies in cognitive science on capabilities that emerge early in childhood development, we define a taxonomy that captures a meaningful subset of commonsense reasoning capabilities useful in manipulation tasks. We consider three categories of commonsense reasoning capabilities: *physical reasoning*, *affordances*, and *object permanence*. Fig. 1 illustrates our taxonomy and representative tasks per category.

Physical reasoning refers to reasoning about object presence and relationships under partial observability, inspired by Renée Baillargeon's work on three key infant event categories [2]: *occlusion*, *covering*, and *containment*. A target object may be surrounded by occluding clutter (occlusion), hidden beneath covering object(s) (covering), or enclosed within a container such as a drawer (containment). While Baillargeon's paradigms probe perceptual expectations about hidden objects, we consider a closed-loop control setting, where the agent must act—not merely predict—under each partial observability condition and reason about avoiding or moving aside distractors to manipulate the target objects.

Next, we consider *affordances*, inspired by James J. Gibson's notion [15] of the actions that an object or environment permits. We consider two key subcategories of affordances. We use the term *spatial affordances* to refer to action possibilities stemming from surface layouts and object geometries; in a block-stacking task, for example, a block only affords supporting blocks its size or smaller, so a policy must reason about stacking blocks in descending order of size. We use *grasping affordances* to refer to an expansion of Gibson's notion within robotics: the actions that specific object parts facilitate [31, 10]. For example, a mug affords grasping by its handle when it must be handed to someone (to avoid touching the part they will drink from). We operationalize affordance reasoning through tasks requiring policies to identify action-relevant object parts or spatial configurations and execute corresponding manipulations.

Finally, we consider *object permanence*, inspired by Jean Piaget's studies [36] on children's ability to interact with objects not in view but previously observed or interacted with. Piaget identified multiple stages of object permanence, ranging in difficulty from finding objects after observing their occlusion to tracking those objects through displacements while occluded. We develop tasks testing Piaget's Stage 4, in which an agent must be capable of recovering an object after observing its occlusion in one of multiple possible locations without tracking displacements.

B. MolmoSpaces-CommonSense Benchmark Construction

We instantiate our taxonomy as a set of eight manipulation tasks distributed across the above categories (see Table I), implemented within the MolmoSpaces ecosystem, which uses the MuJoCo physics simulator. Tasks were selected to be tractable to simulate and reflect realistic manipulation settings. For all tasks, initial scenes per environment are sampled from the MSCrafted [25] and MSProc [8] scene datasets, and we randomize target objects, object poses, exocentric camera poses, and robot base poses. All environments use a DROID [23] setup (i.e., a Franka FR3 with the RobotIQ 2F-85 gripper, along with matching camera intrinsics and extrinsics). We additionally apply a planner-based feasibility filter during procedural generation: candidate environments are retained only if a privileged task-specific planner succeeds (following MolmoBot's [9] data generation pipeline; see Appendix for more details).

Physical reasoning. We instantiate single-object pick and packing tasks for each subcategory. Environment creation entails procedural repositioning of distractor objects: surrounding the target (occlusion), covering it (covering), or enclosing it within an articulated object (containment).

Affordances. We instantiate one task per subcategory. For grasping affordances, we design pick tasks with targeted prompts, e.g. "pick up the hot pan", where success requires not only lifting the object but grasping it at the task-appropriate part (e.g., the pan's handle). For spatial affordances, we design a block-stacking task (with randomized placement of 2-4 blocks in each environment) in which the policy must reason about larger blocks' ability to support smaller ones.

Object Permanence. We instantiate a single task targeting Piaget's Stage 4: each environment initializes two upside-down mugs suspended above a surface with a ball under one, providing a momentary observation window before the mugs fall onto the surface and hide the ball. The policy must subsequently pick the correct mug to reveal the ball.

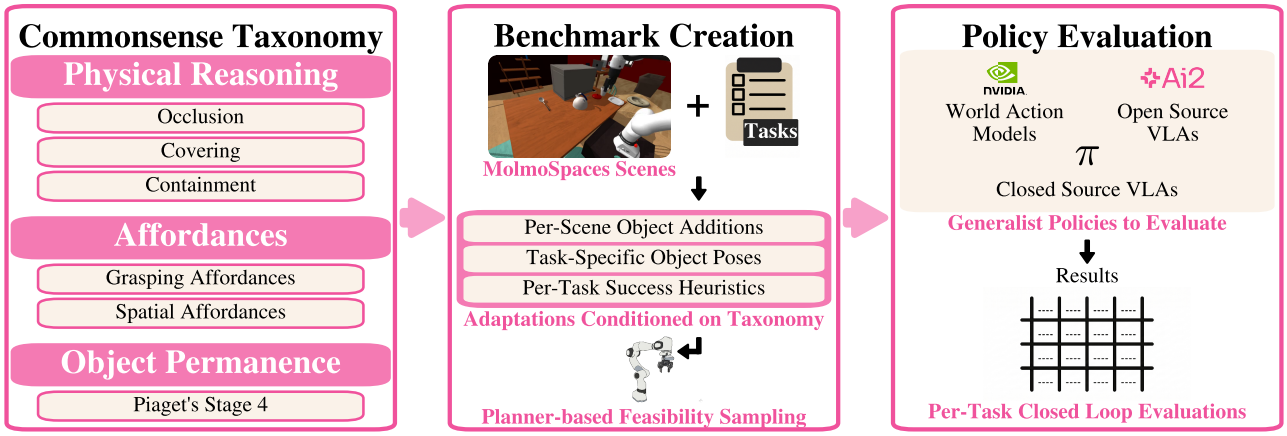

Fig. 2: Our systematic process for analyzing commonsense capabilities of models. Starting from a taxonomy of commonsense tasks, we implement corresponding benchmarks by utilizing the Molmospaces ecosystem and evaluate and analyze diverse policy classes.

TABLE I: Overview of the tasks in our benchmark, MolmoSpaces-CommonSense. Each of the eight tasks evaluates a distinct aspect of commonsense reasoning.

Task	Category	Subcategory	# Envs	Success Condition
Pick with Occlusion	Phys. Reas.	Occlusion	1000	Target object lifted
Pick with Covering	Phys. Reas.	Covering	1000	Target object lifted
Pick with Containment	Phys. Reas.	Containment	500	Target object lifted
Pack with Occlusion	Phys. Reas.	Occlusion	1000	All objects in receptacle
Pack with Covering	Phys. Reas.	Covering	1000	All objects in receptacle
Pick with Affordance	Affordances	Grasping	500	Lifted + correct grasp pose
Stack with Affordance	Affordances	Spatial	1000	All blocks stacked
Pick with Object Permanence	Object Per.	Piaget St. 4	1000	Correct mug will ball underneath lifted

C. Evaluation Protocol

Robot Policy Evaluations. We evaluate policies closed-loop and report per-task success rates with Bayesian beta-binomial credible intervals. Success is checked programmatically in MuJoCo (lift height, receptacle containment via ray-casting, grasp-pose matching against annotations, and block-contact checks); see Appendix for full specifications.

VLM as a Judge Evaluations. We additionally evaluate how well VLMs can *verify* policy rollouts—a strictly easier task than *generating* action trajectories, since the verifier only needs to recognize whether the trajectory succeeded, not produce one. This partially isolates perception and reasoning from action. For the Pick with Occlusion task, we annotate a sample of rollouts from representative policies with one of five outcome categories (four failure modes plus success; see Appendix), then prompt several VLMs to classify each rollout video and measure accuracy against these annotations.

IV. Experimental Results

To begin, we report success rates of various generalist policies when evaluated zero-shot on all tasks within MolmoSpaces-Commonsense (see Table II). Our evaluation spans policies with diverse architectures: $\pi_{0.5}$, π_0-FAST, and π_0 (all VLAs), MolmoAct2 (VLA with depth token reasoning traces), and DreamZero (WAM). Notably, no policy achieves a success rate over 30% on any of our tasks. Our distributional analyses further reveal that *these architecturally distinct policies fail in highly correlated ways*, pointing to a bottleneck in the shared data and training paradigm rather than architecture.

More broadly, we find that across all taxonomy categories, policies fall well short of capable performance. Within physical reasoning tasks, success drops below 20% under occlusion or covering and to 0% under containment. Policies struggle to recognize and act upon grasping and spatial affordances (with success on the latter being near zero). Additionally, no policy succeeds on object permanence due to inability to handle observation history.

We scope distributional analyses to tasks with non-zero zero-shot performance. Containment is excluded as uninformative and plausibly out-of-distribution, and object permanence is excluded due to the above structural concerns with existing generalist policies. We leave both to future work. Our VLM evaluations use Gemini 3.1 [16], Gemini Robotics ER 1.6 [17], Qwen3.5 [38], and Molmo2 [6].

TABLE II: Success rates (%) across policies and tasks, showing policies struggle across all tasks. We report the mean and 95% Bayesian credible interval using a Beta(1,1) prior. Best per column (among generalist policies) in **bold**.

Policy	*Physical Reasoning*					*Affordance*		*Obj. Perm.*
	Pick w/ Occlusion	Pick w/ Covering	Pick w/ Containment	Pack w/ Occlusion	Pack w/ Covering	Pick w/ Affordance	Stack w/ Affordance	Pick w/ Obj. Permanence
π_0	$2.81_{\pm 1.22}$	$2.80_{\pm 1.22}$	$0.00_{\pm 0.73}$	$3.30_{\pm 1.30}$	$4.30_{\pm 1.44}$	$9.40_{\pm 2.88}$	$0.10_{\pm 0.46}$	$0.00_{\pm 0.37}$
π_0-FAST	$8.21_{\pm 1.87}$	$6.91_{\pm 1.74}$	$0.00_{\pm 0.73}$	$2.20_{\pm 1.11}$	$5.60_{\pm 1.60}$	$13.40_{\pm 3.27}$	$\mathbf{0.20_{\pm 0.52}}$	$0.00_{\pm 0.37}$
$\pi_{0.5}$	$14.10_{\pm 2.30}$	$11.31_{\pm 2.02}$	$0.00_{\pm 0.73}$	$\mathbf{12.10_{\pm 2.17}}$	$\mathbf{14.20_{\pm 2.30}}$	$19.40_{\pm 3.70}$	$\mathbf{0.20_{\pm 0.52}}$	$\mathbf{0.30_{\pm 0.57}}$
DreamZero	$\mathbf{18.00_{\pm 2.50}}$	$13.80_{\pm 2.28}$	$0.00_{\pm 0.73}$	$7.00_{\pm 1.75}$	$7.10_{\pm 1.76}$	$\mathbf{26.60_{\pm 4.04}}$	$0.10_{\pm 0.46}$	$\mathbf{0.30_{\pm 0.57}}$
MolmoAct2	$17.20_{\pm 2.47}$	$\mathbf{14.00_{\pm 2.29}}$	$0.00_{\pm 0.73}$	$8.96_{\pm 1.94}$	$10.14_{\pm 2.03}$	$23.60_{\pm 3.92}$	$\mathbf{0.20_{\pm 0.52}}$	$0.00_{\pm 0.37}$
Planner	100.00	100.00	100.00	100.00	100.00	100.00	100.00	100.00

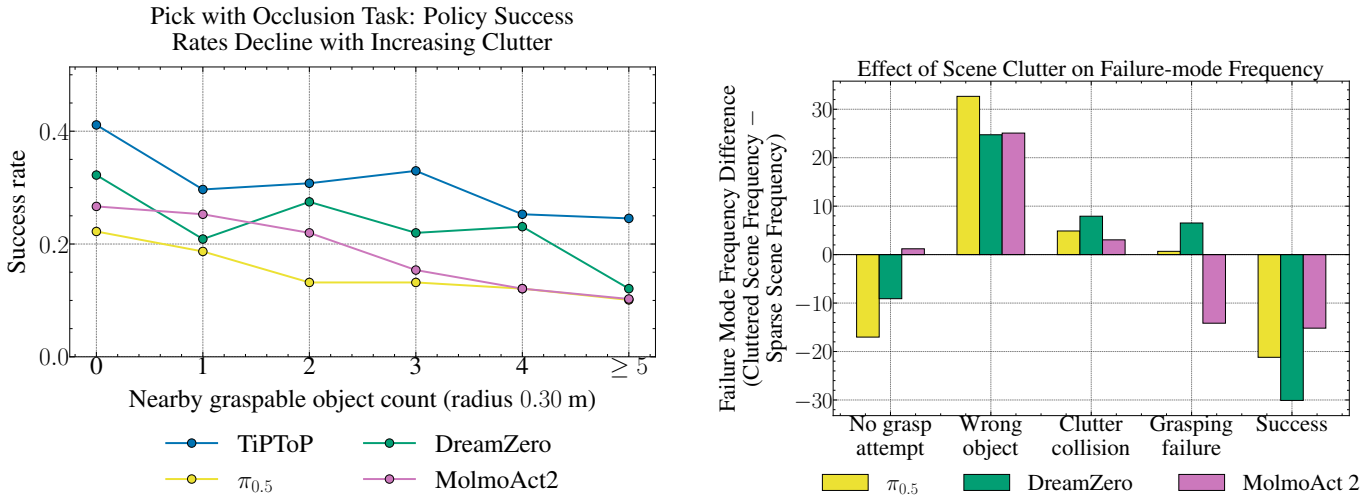

Fig. 3: Pick-with-occlusion task analysis. **(Left)** Policy success rates binned by the number of distractor objects within a small radius of the target pick object. **(Right)** Change in occurrence of failure modes as we go from sparse (≤ 1 nearby objects) to cluttered scenes (> 1 nearby objects)

A. How Does Policy Performance Change with Clutter?

In our physical reasoning tasks, success rates degrade consistently with scene clutter across all evaluated architectures. Fig. 3 visualizes this in more detail for our Pick with Occlusion task, showing a near monotonic decline in success rate as the number of distractor objects near the target pick object increase. Additionally, success rates across all policies fall by over 50% with 5+ distractor objects, relative to the zero-distractor setting. For this task, we additionally evaluate TiPToP [42], a VLM-enabled modular planner (not a generalist policy). TiPToP reaches nearly double the performance of the best generalist policy, but also degrades significantly with clutter, indicating the difficulty is not specific to end-to-end policies.

Additionally, for a random sample of $n = 200$ scenes from the pick occlusion task, we evaluate different failure modes (see Fig. 3). After binning episodes by scene clutter (as measured by number of nearby distractor objects), we observe that in the high clutter subset, approaching/grasping the incorrect distractor object becomes a dominant failure mode, accounting almost entirely for the drop in success rate and indicating that perception is a weak link.

> **Takeaway 1:** Clutter significantly reduces current policies' performance across architectures, motivating data collection and evaluation in more complex environments going forward.

B. Are Policies Steerable by Language?

On our Pick with Affordance task, we see that no policy exceeds a 30% success rate. We address the possible explanation that these policies are simply poor at grasping in general by reporting success rates within the same environments but without the grasping location success condition (see Fig. 4). Across all policies, success rates are nearly halved with the additional grasp location check. This points to failures specifically in reasoning about grasping affordances.

Additionally, we ablate the choice of prompt used in this task. Examples of our standard prompts include "pick up the hot pan" or "pick up the mug to give to someone", and we

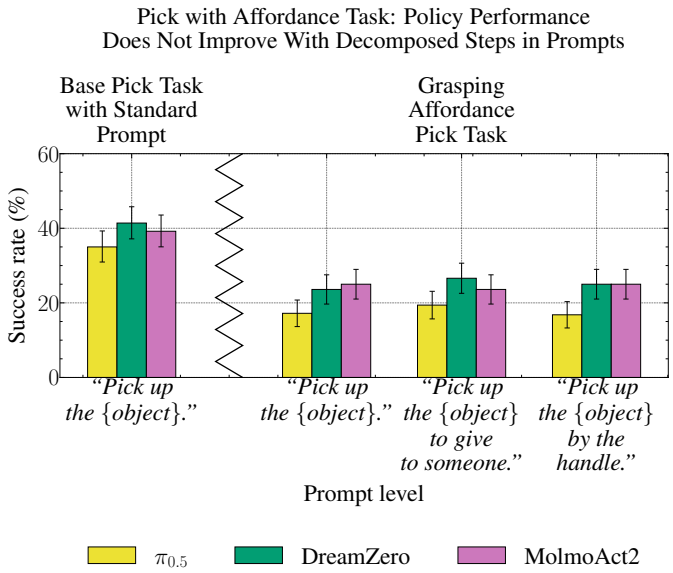

Fig. 4: Success rates for base pick task (left), side by side with our Pick with Affordance task (right). We ablate multiple levels of prompt detail.

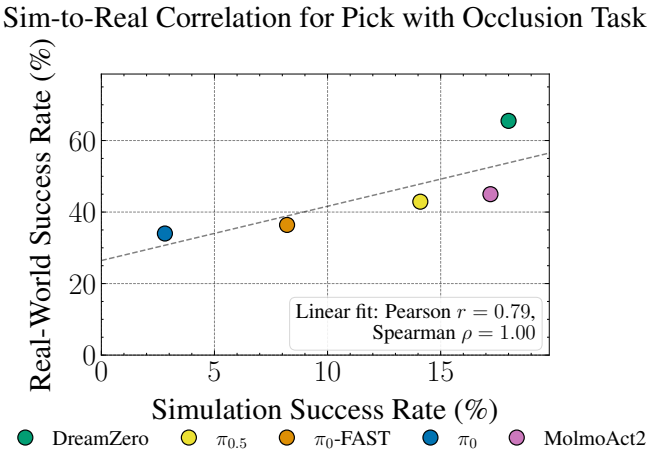

Fig. 5: Sim-to-real correlation results for our Pick with Occlusion task.

test both more specific instructions ("pick up the pan by the handle") and the base task with no steering ("pick up the pan"). We find minimal impact from language steering across policies in either direction; neither added specificity nor its removal significantly shifts success rates.

Takeaway 2: Policies show limited steerability with respect to specified affordances, suggesting that training datasets should pair precise language with the behaviors those instructions describe.

C. Sim-to-Real Correlation

Cluttered scenes are a significant new evaluation setting MolmoSpaces-Commonsense introduces over MolmoSpaces, which already establishes strong sim-to-real correlations for atomic skills (pick, open, close). To verify sim-to-real correlation in this regime, we run in-house real-world rollouts of MolmoAct 2 on pick-with-occlusion and supplement these with filtered episodes from RoboArena [1], retaining medium-to-high clutter scenes — 295 total real episodes against 1000 simulation episodes per policy. Rank order is preserved across the five evaluated policies (Fig. 5), with reasonable correlation ($r = 0.79$) between simulation and real success rates.

TABLE III: VLM-as-judge accuracy (%) for classifying rollout outcomes across three VLA policies ($n = 200$ rollouts per policy). Across the board, VLMs struggle to determine why a policy succeeded or failed in a given trajectory.

Evaluator	$\pi_{0.5}$	DreamZero	MolmoAct2	Mean
Molmo2	27.9	37.4	19.4	28.2
Qwen3.5-122B-A10B	44.6	46.1	44.8	45.2
Gemini 3.1	**46.6**	**63.1**	43.8	51.2
Gemini Robotics-ER 1.6	50.5	51.0	**53.2**	**51.6**

D. Can VLMs Judge Policy Rollouts from MolmoSpaces-CommonSense?

On the perception-only VLM judge task, no model exceeds a performance of ∼63% (refer to Table III for full results), indicating the perception and reasoning task alone is challenging and not just action prediction. We additionally notice common failure modes with rollout classification. In settings where the gripper does not make contact with any objects in the scene, models mistakenly output that the policy made contact and attempted to grasp an object (with some models, like Gemini 3.1, this happens more than 50% of the time). For successful trajectories, VLMs often misclassify these as grasping failures (notably, Gemini Robotics ER 1.6 does this over 50% of the time for DreamZero rollouts).

Takeaway 3: VLMs struggle to judge trajectories, showing MolmoSpaces-CommonSense poses difficulties across the full stack of perception, reasoning, and action. This indicates that modular systems based on these models may still struggle on our tasks.

V. Discussion

A notable pattern in our analysis is that architecturally distinct policies (VLAs, WAMs) fail in highly correlated ways: all evaluated policies deteriorate similarly under clutter, with wrong-object selection as the dominant failure mode, and show shared insensitivity to language steering. This pattern suggests that gaps exist upstream of architecture, in the existing training datasets and evaluation procedures used for robot policies, and we hope that the simulation environments in MolmoSpaces-CommonSense can be used to address these gaps. Our analysis also shows that the difficulty of tasks in our commonsense taxonomy extends to all parts of the perception-reasoning-action stack.

Limitations and Future Work. We focus on evaluating policies zero-shot, which limits our scope of analysis to the subset of our taxonomy on which these policies attain a threshold success. In future work, we plan to leverage these environments to train policies, analyzing existing shortcomings and uncovering improvements for commonsense reasoning. As generalist policies improve, we also hope to augment our set of tasks to cover more advanced capabilities within our commonsense taxonomy (e.g. higher stages of object permanence understanding), as well as additional categories of commonsense reasoning (e.g. intuitive physics). We additionally caveat our VLM-judge experiments: the judge is a model distinct from the policies being evaluated, so it bounds rather than pinpoints the specific reasoning failures of the policies themselves. We treat the experiment as analytically useful nonetheless — verifying a rollout draws on perceptual and reasoning capabilities similar to those required to solve the task.

Conclusion. We present MolmoSpaces-CommonSense, a benchmark that tests the combination of commonsense reasoning and action for generalist policies. Our goal is to facilitate a new line of inquiry into building robot policies that can not only execute atomic skills but also reason through the commonsense-laden situations the real world presents.

References

[1] Pranav Atreya, Karl Pertsch, Tony Lee, Moo Jin Kim, Arhan Jain, Artur Kuramshin, Clemens Eppner, Cyrus Neary, Edward Hu, Fabio Ramos, Jonathan Tremblay, Kanav Arora, Kirsty Ellis, Luca Macesanu, Marcel Torne Villasevil, Matthew Leonard, Meedeum Cho, Ozgur Aslan, Shivin Dass, Jie Wang, William Reger, Xingfang Yuan, Xuning Yang, Abhishek Gupta, Dinesh Jayaraman, Glen Berseth, Kostas Daniilidis, Roberto Martin-Martin, Youngwoon Lee, Percy Liang, Chelsea Finn, and Sergey Levine. Roboarena: Distributed real-world evaluation of generalist robot policies, 2025. URL https://arxiv.org/abs/2506.18123.

[2] Renée Baillargeon. Infants' physical world. *Current Directions in Psychological Science*, 13(3):89–94, 2004. doi: 10.1111/j.0963-7214.2004.00281.x.

[3] 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, October 2024. URL http://arxiv.org/abs/2410.24164. arXiv:2410.24164 [cs].

[4] Boyuan Chen, Zhuo Xu, Sean Kirmani, Brian Ichter, Danny Driess, Pete Florence, Dorsa Sadigh, Leonidas Guibas, and Fei Xia. SpatialVLM: Endowing Vision-Language Models with Spatial Reasoning Capabilities, January 2024. URL http://arxiv.org/abs/2401.12168. arXiv:2401.12168 [cs].

[5] Long Cheng, Jiafei Duan, Yi Ru Wang, Haoquan Fang, Boyang Li, Yushan Huang, Elvis Wang, Ainaz Eftekhar, Jason Lee, Wentao Yuan, Rose Hendrix, Noah A. Smith, Fei Xia, Dieter Fox, and Ranjay Krishna. Pointarena: Probing multimodal grounding through language-guided pointing. *arXiv preprint arXiv:2505.09990*, 2025.

[6] Christopher Clark, Jieyu Zhang, Zixian Ma, Jae Sung Park, Mohammadreza Salehi, Rohun Tripathi, Sangho Lee, Zhongzheng Ren, Chris Dongjoo Kim, Yinuo Yang, Vincent Shao, Yue Yang, Weikai Huang, Ziqi Gao, Taira Anderson, Jianrui Zhang, Jitesh Jain, George Stoica, Winson Han, Ali Farhadi, and Ranjay Krishna. Molmo2: Open weights and data for vision-language models with video understanding and grounding, 2026. URL https://arxiv.org/abs/2601.10611.

[7] Yinpei Dai, Hongze Fu, Jayjun Lee, Yuejiang Liu, Haoran Zhang, Jianing Yang, Chelsea Finn, Nima Fazeli, and Joyce Chai. RoboMME: Benchmarking and Understanding Memory for Robotic Generalist Policies, March 2026. URL http://arxiv.org/abs/2603.04639. arXiv:2603.04639 [cs].

[8] Matt Deitke, Eli VanderBilt, Alvaro Herrasti, Luca Weihs, Jordi Salvador, Kiana Ehsani, Winson Han, Eric Kolve, Ali Farhadi, Aniruddha Kembhavi, and Roozbeh Mottaghi. ProcTHOR: Large-Scale Embodied AI Using Procedural Generation, June 2022. URL http://arxiv.org/abs/2206.06994. arXiv:2206.06994 [cs].

[9] Abhay Deshpande, Maya Guru, Rose Hendrix, Snehal Jauhri, Ainaz Eftekhar, Rohun Tripathi, Max Argus, Jordi Salvador, Haoquan Fang, Matthew Wallingford, Wilbert Pumacay, Yejin Kim, Quinn Pfeifer, Ying-Chun Lee, Piper Wolters, Omar Rayyan, Mingtong Zhang, Jiafei Duan, Karen Farley, Winson Han, Eli Vanderbilt, Dieter Fox, Ali Farhadi, Georgia Chalvatzaki, Dhruv Shah, and Ranjay Krishna. MolmoBot: Large-scale simulation enables zero-shot manipulation, 2026. URL https://arxiv.org/abs/2603.16861. Technical Report, Allen Institute for AI.

[10] Thanh-Toan Do, Anh Nguyen, and Ian Reid. AffordanceNet: An end-to-end deep learning approach for object affordance detection. In *2018 IEEE International Conference on Robotics and Automation (ICRA)*, pages 5882–5889. IEEE, 2018.

[11] Haoquan Fang, Markus Grotz, Wilbert Pumacay, Yi Ru Wang, Dieter Fox, Ranjay Krishna, and Jiafei Duan. SAM2Act: Integrating Visual Foundation Model with A Memory Architecture for Robotic Manipulation, July 2025. URL http://arxiv.org/abs/2501.18564. arXiv:2501.18564 [cs].

[12] Haoquan Fang, Jiafei Duan, Donovan Clay, Sam Wang, Shuo Liu, Weikai Huang, Xiang Fan, Wei-Chuan Tsai, Shirui Chen, Yi Ru Wang, Shanli Xing, Jaemin Cho, Jae Sung Park, Ainaz Eftekhar, Peter Sushko, Karen Farley, Angad Wadhwa, Cole Harrison, Winson Han,

Ying-Chun Lee, Eli VanderBilt, Rose Hendrix, Suveen Ellawela, Lucas Ngoo, Joyce Chai, Zhongzheng Ren, Ali Farhadi, Dieter Fox, and Ranjay Krishna. MolmoAct2: Action reasoning models for real-world deployment. Allen Institute for AI, 2026. URL https://allenai.org/papers/molmoact2.

[13] Senyu Fei, Siyin Wang, Junhao Shi, Zihao Dai, Jikun Cai, Pengfang Qian, Li Ji, Xinzhe He, Shiduo Zhang, Zhaoye Fei, Jinlan Fu, Jingjing Gong, and Xipeng Qiu. LIBERO-Plus: In-depth Robustness Analysis of Vision-Language-Action Models, December 2025. URL http://arxiv.org/abs/2510.13626. arXiv:2510.13626 [cs].

[14] Jensen Gao, Suneel Belkhale, Sudeep Dasari, Ashwin Balakrishna, Dhruv Shah, and Dorsa Sadigh. A Taxonomy for Evaluating Generalist Robot Manipulation Policies, February 2026. URL http://arxiv.org/abs/2503.01238. arXiv:2503.01238 [cs].

[15] James J. Gibson. *The Ecological Approach to Visual Perception*. Houghton Mifflin, Boston, 1979. ISBN 978-0395270493.

[16] Google DeepMind. Gemini 3.1 pro model card. https://deepmind.google/models/model-cards/gemini-3-1-pro/, 2026. Accessed 2026-05-28.

[17] Google DeepMind. Gemini robotics-er 1.6: Powering real-world robotics tasks through enhanced embodied reasoning. https://deepmind.google/blog/gemini-robotics-er-1-6/, 2026. Accessed 2026-05-28.

[18] Xiaoshuai Hao, Yingbo Tang, Lingfeng Zhang, Yanbiao Ma, Yunfeng Diao, Ziyu Jia, Wenbo Ding, Hangjun Ye, and Long Chen. RoboAfford++: A Generative AI-Enhanced Dataset for Multimodal Affordance Learning in Robotic Manipulation and Navigation, 2025. URL https://arxiv.org/abs/2511.12436.

[19] Jack Hessel, Jena D. Hwang, Jae Sung Park, Rowan Zellers, Chandra Bhagavatula, Anna Rohrbach, Kate Saenko, and Yejin Choi. The Abduction of Sherlock Holmes: A Dataset for Visual Abductive Reasoning, July 2022. URL http://arxiv.org/abs/2202.04800. arXiv:2202.04800 [cs].

[20] Drew A. Hudson and Christopher D. Manning. GQA: A New Dataset for Real-World Visual Reasoning and Compositional Question Answering, May 2019. URL http://arxiv.org/abs/1902.09506. arXiv:1902.09506 [cs].

[21] 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, April 2025. URL http://arxiv.org/abs/2504.16054. arXiv:2504.16054 [cs].

[22] Stephen James, Zicong Ma, David Rovick Arrojo, and Andrew J. Davison. RLBench: The Robot Learning Benchmark & Learning Environment, September 2019. URL http://arxiv.org/abs/1909.12271. arXiv:1909.12271 [cs].

[23] Alexander Khazatsky, Karl Pertsch, Suraj Nair, Ashwin Balakrishna, Sudeep Dasari, Siddharth Karamcheti, Soroush Nasiriany, Mohan Kumar Srirama, Lawrence Yunliang Chen, Kirsty Ellis, Peter David Fagan, Joey Hejna, Masha Itkina, Marion Lepert, Yecheng Jason Ma, Patrick Tree Miller, Jimmy Wu, Suneel Belkhale, Shivin Dass, Huy Ha, Arhan Jain, Abraham Lee, Youngwoon Lee, Marius Memmel, Sungjae Park, Ilija Radosavovic, Kaiyuan Wang, Albert Zhan, Kevin Black, Cheng Chi, Kyle Beltran Hatch, Shan Lin, Jingpei Lu, Jean Mercat, Abdul Rehman, Pannag R. Sanketi, Archit Sharma, Cody Simpson, Quan Vuong, Homer Rich Walke, Blake Wulfe, Ted Xiao, Jonathan Heewon Yang, Arefeh Yavary, Tony Z. Zhao, Christopher Agia, Rohan Baijal, Mateo Guaman Castro, Daphne Chen, Qiuyu Chen, Trinity Chung, Jaimyn Drake, Ethan Paul Foster, Jensen Gao, Vitor Guizilini, David Antonio Herrera, Minho Heo, Kyle Hsu, Jiaheng Hu, Muhammad Zubair Irshad, Donovon Jackson, Charlotte Le, Yunshuang Li, Kevin Lin, Roy Lin, Zehan Ma, Abhiram Maddukuri, Suvir Mirchandani, Daniel Morton, Tony Nguyen, Abigail O'Neill, Rosario Scalise, Derick Seale, Victor Son, Stephen Tian, Emi Tran, Andrew E. Wang, Yilin Wu, Annie Xie, Jingyun Yang, Patrick Yin, Yunchu Zhang, Osbert Bastani, Glen Berseth, Jeannette Bohg, Ken Goldberg, Abhinav Gupta, Abhishek Gupta, Dinesh Jayaraman, Joseph J. Lim, Jitendra Malik, Roberto Martín-Martín, Subramanian Ramamoorthy, Dorsa Sadigh, Shuran Song, Jiajun Wu, Michael C. Yip, Yuke Zhu, Thomas Kollar, Sergey Levine, and Chelsea Finn. DROID: A Large-Scale In-The-Wild Robot Manipulation Dataset, April 2025. URL http://arxiv.org/abs/2403.12945. arXiv:2403.12945 [cs].

[24] Yejin Kim, Wilbert Pumacay, Omar Rayyan, Max Argus, Winson Han, Eli VanderBilt, Jordi Salvador, Abhay Deshpande, Rose Hendrix, Snehal Jauhri, Shuo Liu, Nur Muhammad Mahi Shafiullah, Maya Guru, Ainaz Eftekhar, Karen Farley, Donovan Clay, Jiafei Duan, Arjun Guru, Piper Wolters, Alvaro Herrasti, Ying-Chun Lee, Georgia Chalvatzaki, Yuchen Cui, Ali Farhadi, Dieter Fox, and Ranjay Krishna. MolmoSpaces: A Large-Scale Open Ecosystem for Robot Navigation and Manipulation, February 2026. URL http://arxiv.org/abs/2602.11337. arXiv:2602.11337 [cs].

[25] Eric Kolve, Roozbeh Mottaghi, Winson Han, Eli VanderBilt, Luca Weihs, Alvaro Herrasti, Matt Deitke, Kiana Ehsani, Daniel Gordon, Yuke Zhu, Aniruddha Kembhavi, Abhinav Gupta, and Ali Farhadi. AI2-THOR: An Interactive 3D Environment for Visual AI, August 2022. URL http://arxiv.org/abs/1712.05474. arXiv:1712.05474 [cs].

[26] Chengshu Li, Ruohan Zhang, Josiah Wong, Cem Gokmen, Sanjana Srivastava, Roberto Martín-Martín, Chen Wang, Gabrael Levine, Wensi Ai, Benjamin Martinez, Hang Yin, Michael Lingelbach, Minjune Hwang, Ayano Hiranaka, Sujay Garlanka, Arman Aydin, Sharon Lee, Jiankai Sun, Mona Anvari, Manasi Sharma, Dhruva Bansal, Samuel Hunter, Kyu-Young Kim, Alan Lou, Caleb R. Matthews, Ivan Villa-Renteria, Jerry Huayang Tang, Claire Tang, Fei Xia, Yunzhu Li, Silvio Savarese, Hyowon Gweon, C. Karen Liu, Jiajun Wu, and Li Fei-Fei. BEHAVIOR-1K: A Human-Centered, Embodied AI Benchmark with 1,000 Everyday Activities and Realistic Simulation, March 2024. URL http://arxiv.org/abs/2403.09227. arXiv:2403.09227 [cs].

[27] Xuanlin Li, Kyle Hsu, Jiayuan Gu, Karl Pertsch, Oier Mees, 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, May 2024. URL http://arxiv.org/abs/2405.05941. arXiv:2405.05941 [cs].

[28] Bo Liu, Yifeng Zhu, Chongkai Gao, Yihao Feng, Qiang Liu, Yuke Zhu, and Peter Stone. LIBERO: Benchmarking Knowledge Transfer for Lifelong Robot Learning, October 2023. URL http://arxiv.org/abs/2306.03310. arXiv:2306.03310 [cs].

[29] Kenneth Marino, Mohammad Rastegari, Ali Farhadi, and Roozbeh Mottaghi. OK-VQA: A Visual Question Answering Benchmark Requiring External Knowledge, September 2019. URL http://arxiv.org/abs/1906.00067. arXiv:1906.00067 [cs].

[30] Oier Mees, Lukas Hermann, Erick Rosete-Beas, and Wolfram Burgard. CALVIN: A Benchmark for Language-Conditioned Policy Learning for Long-Horizon Robot Manipulation Tasks, July 2022. URL http://arxiv.org/abs/2112.03227. arXiv:2112.03227 [cs].

[31] Austin Myers, Ching L. Teo, Cornelia Fermüller, and Yiannis Aloimonos. Affordance detection of tool parts from geometric features. In *2015 IEEE International Conference on Robotics and Automation (ICRA)*, pages 1374–1381. IEEE, 2015.

[32] Soroush Nasiriany, Abhiram Maddukuri, Lance Zhang, Adeet Parikh, Aaron Lo, Abhishek Joshi, Ajay Mandlekar, and Yuke Zhu. RoboCasa: Large-Scale Simulation of Everyday Tasks for Generalist Robots, June 2024. URL http://arxiv.org/abs/2406.02523. arXiv:2406.02523 [cs].

[33] Soroush Nasiriany, Sepehr Nasiriany, Abhiram Maddukuri, and Yuke Zhu. RoboCasa365: A Large-Scale Simulation Framework for Training and Benchmarking Generalist Robots, March 2026. URL http://arxiv.org/abs/2603.04356. arXiv:2603.04356 [cs].

[34] Jae Sung Park, Chandra Bhagavatula, Roozbeh Mottaghi, Ali Farhadi, and Yejin Choi. VisualCOMET: Reasoning about the Dynamic Context of a Still Image, August 2020. URL http://arxiv.org/abs/2004.10796. arXiv:2004.10796 [cs].

[35] 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, January 2025. URL http://arxiv.org/abs/2501.09747. arXiv:2501.09747 [cs].

[36] Jean Piaget. *The Construction of Reality in the Child*. Basic Books, New York, 1954. Originally published as *La Construction du Réel chez l'Enfant*, 1937.

[37] 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 XX*. Robotics: Science and Systems Foundation, July 2024. ISBN 979-8-9902848-0-7. doi: 10.15607/RSS.2024.XX.133. URL http://www.roboticsproceedings.org/rss20/p133.pdf.

[38] Qwen Team. Qwen3.5. https://huggingface.co/Qwen/Qwen3.5-397B-A17B, 2026. Apache-2.0 vision-language foundation models; Hybrid Gated DeltaNet + sparse MoE. Variants incl. 35B-A3B, 122B-A10B, 397B-A17B. Released 2026-02-16. Accessed 2026-05-28.

[39] Ronan Riochet, Mario Ynocente Castro, Mathieu Bernard, Adam Lerer, Rob Fergus, Véronique Izard, and Emmanuel Dupoux. IntPhys: A Framework and Benchmark for Visual Intuitive Physics Reasoning, February 2020. URL http://arxiv.org/abs/1803.07616. arXiv:1803.07616 [cs].

[40] Keisuke Sakaguchi, Ronan Le Bras, Chandra Bhagavatula, and Yejin Choi. WinoGrande: An Adversarial Winograd Schema Challenge at Scale, November 2019. URL http://arxiv.org/abs/1907.10641. arXiv:1907.10641 [cs].

[41] Maarten Sap, Ronan LeBras, Emily Allaway, Chandra Bhagavatula, Nicholas Lourie, Hannah Rashkin, Brendan Roof, Noah A. Smith, and Yejin Choi. ATOMIC: An Atlas of Machine Commonsense for If-Then Reasoning, February 2019. URL http://arxiv.org/abs/1811.00146. arXiv:1811.00146 [cs].

[42] William Shen, Nishanth Kumar, Sahit Chintalapudi, Jie Wang, Christopher Watson, Edward Hu, Jing Cao, Dinesh Jayaraman, Leslie Pack Kaelbling, and Tomás Lozano-Pérez. Tiptop: A modular open-vocabulary planning system for robotic manipulation, 2026. URL https://arxiv.org/abs/2603.09971.

[43] Chan Hee Song, Valts Blukis, Jonathan Tremblay, Stephen Tyree, Yu Su, and Stan Birchfield. RoboSpatial: Teaching Spatial Understanding to 2D and 3D Vision-Language Models for Robotics, February 2026. URL http://arxiv.org/abs/2411.16537. arXiv:2411.16537 [cs].

[44] Gemini Robotics Team, Abbas Abdolmaleki, Saminda Abeyruwan, Joshua Ainslie, Jean-Baptiste Alayrac, Montserrat Gonzalez Arenas, Ashwin Balakrishna, Nathan Batchelor, Alex Bewley, Jeff Bingham, Michael Bloesch, Konstantinos Bousmalis, Philemon Brakel, An-

thony Brohan, Thomas Buschmann, Arunkumar Byravan, Serkan Cabi, Ken Caluwaerts, Federico Casarini, Christine Chan, Oscar Chang, London Chappellet-Volpini, Jose Enrique Chen, Xi Chen, Hao-Tien Lewis Chiang, Krzysztof Choromanski, Adrian Collister, David B. D'Ambrosio, Sudeep Dasari, Todor Davchev, Meet Kirankumar Dave, Coline Devin, Norman Di Palo, Tianli Ding, Carl Doersch, Adil Dostmohamed, Yilun Du, Debidatta Dwibedi, Sathish Thoppay Egambaram, Michael Elabd, Tom Erez, Xiaolin Fang, Claudio Fantacci, Cody Fong, Erik Frey, Chuyuan Fu, Ruiqi Gao, Marissa Giustina, Keerthana Gopalakrishnan, Laura Graesser, Oliver Groth, Agrim Gupta, Roland Hafner, Steven Hansen, Leonard Hasenclever, Sam Haves, Nicolas Heess, Brandon Hernaez, Alex Hofer, Jasmine Hsu, Lu Huang, Sandy H. Huang, Atil Iscen, Mithun George Jacob, Deepali Jain, Sally Jesmonth, Abhishek Jindal, Ryan Julian, Dmitry Kalashnikov, M. Emre Karagozler, Stefani Karp, Matija Kecman, J. Chase Kew, Donnie Kim, Frank Kim, Junkyung Kim, Thomas Kipf, Sean Kirmani, Ksenia Konyushkova, Li Yang Ku, Yuheng Kuang, Thomas Lampe, Antoine Laurens, Tuan Anh Le, Isabel Leal, Alex X. Lee, Tsang-Wei Edward Lee, Guy Lever, Jacky Liang, Li-Heng Lin, Fangchen Liu, Shangbang Long, Caden Lu, Sharath Maddineni, Anirudha Majumdar, Kevis-Kokitsi Maninis, Andrew Marmon, Sergio Martinez, Assaf Hurwitz Michaely, Niko Milonopoulos, Joss Moore, Robert Moreno, Michael Neunert, Francesco Nori, Joy Ortiz, Kenneth Oslund, Carolina Parada, Emilio Parisotto, Amaris Paryag, Acorn Pooley, Thomas Power, Alessio Quaglino, Haroon Qureshi, Rajkumar Vasudeva Raju, Helen Ran, Dushyant Rao, Kanishka Rao, Isaac Reid, David Rendleman, Krista Reymann, Miguel Rivas, Francesco Romano, Yulia Rubanova, Peter Pastor Sampedro, Pannag R. Sanketi, Dhruv Shah, Mohit Sharma, Kathryn Shea, Mohit Shridhar, Charles Shu, Vikas Sindhwani, Sumeet Singh, Radu Soricut, Rachel Sterneck, Ian Storz, Razvan Surdulescu, Jie Tan, Jonathan Tompson, Saran Tunyasuvunakool, Jake Varley, Grace Vesom, Giulia Vezzani, Maria Bauza Villalonga, Oriol Vinyals, René Wagner, Ayzaan Wahid, Stefan Welker, Paul Wohlhart, Chengda Wu, Markus Wulfmeier, Fei Xia, Ted Xiao, Annie Xie, Jinyu Xie, Peng Xu, Sichun Xu, Ying Xu, Zhuo Xu, Jimmy Yan, Sherry Yang, Skye Yang, Yuxiang Yang, Hiu Hong Yu, Wenhao Yu, Wentao Yuan, Yuan Yuan, Jingwei Zhang, Tingnan Zhang, Zhiyuan Zhang, Allan Zhou, Guangyao Zhou, and Yuxiang Zhou. Gemini Robotics 1.5: Pushing the Frontier of Generalist Robots with Advanced Embodied Reasoning, Thinking, and Motion Transfer, October 2025. URL http://arxiv.org/abs/2510.03342. arXiv:2510.03342 [cs].

[45] Gemini Robotics Team, Saminda Abeyruwan, Joshua Ainslie, Jean-Baptiste Alayrac, Montserrat Gonzalez Arenas, Travis Armstrong, Ashwin Balakrishna, Robert Baruch, Maria Bauza, Michiel Blokzijl, Steven Bohez, Konstantinos Bousmalis, Anthony Brohan, Thomas Buschmann, Arunkumar Byravan, Serkan Cabi, Ken Caluwaerts, Federico Casarini, Oscar Chang, Jose Enrique Chen, Xi Chen, Hao-Tien Lewis Chiang, Krzysztof Choromanski, David D'Ambrosio, Sudeep Dasari, Todor Davchev, Coline Devin, Norman Di Palo, Tianli Ding, Adil Dostmohamed, Danny Driess, Yilun Du, Debidatta Dwibedi, Michael Elabd, Claudio Fantacci, Cody Fong, Erik Frey, Chuyuan Fu, Marissa Giustina, Keerthana Gopalakrishnan, Laura Graesser, Leonard Hasenclever, Nicolas Heess, Brandon Hernaez, Alexander Herzog, R. Alex Hofer, Jan Humplik, Atil Iscen, Mithun George Jacob, Deepali Jain, Ryan Julian, Dmitry Kalashnikov, M. Emre Karagozler, Stefani Karp, Chase Kew, Jerad Kirkland, Sean Kirmani, Yuheng Kuang, Thomas Lampe, Antoine Laurens, Isabel Leal, Alex X. Lee, Tsang-Wei Edward Lee, Jacky Liang, Yixin Lin, Sharath Maddineni, Anirudha Majumdar, Assaf Hurwitz Michaely, Robert Moreno, Michael Neunert, Francesco Nori, Carolina Parada, Emilio Parisotto, Peter Pastor, Acorn Pooley, Kanishka Rao, Krista Reymann, Dorsa Sadigh, Stefano Saliceti, Pannag Sanketi, Pierre Sermanet, Dhruv Shah, Mohit Sharma, Kathryn Shea, Charles Shu, Vikas Sindhwani, Sumeet Singh, Radu Soricut, Jost Tobias Springenberg, Rachel Sterneck, Razvan Surdulescu, Jie Tan, Jonathan Tompson, Vincent Vanhoucke, Jake Varley, Grace Vesom, Giulia Vezzani, Oriol Vinyals, Ayzaan Wahid, Stefan Welker, Paul Wohlhart, Fei Xia, Ted Xiao, Annie Xie, Jinyu Xie, Peng Xu, Sichun Xu, Ying Xu, Zhuo Xu, Yuxiang Yang, Rui Yao, Sergey Yaroshenko, Wenhao Yu, Wentao Yuan, Jingwei Zhang, Tingnan Zhang, Allan Zhou, and Yuxiang Zhou. Gemini Robotics: Bringing AI into the Physical World, March 2025. URL http://arxiv.org/abs/2503.20020. arXiv:2503.20020 [cs].

[46] Yue Yang, Fan-Yun Sun, Luca Weihs, Eli VanderBilt, Alvaro Herrasti, Winson Han, Jiajun Wu, Nick Haber, Ranjay Krishna, Lingjie Liu, Chris Callison-Burch, Mark Yatskar, Aniruddha Kembhavi, and Christopher Clark. Holodeck: Language Guided Generation of 3D Embodied AI Environments, April 2024. URL http://arxiv.org/abs/2312.09067. arXiv:2312.09067 [cs].

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

[48] Wentao Yuan, Jiafei Duan, Valts Blukis, Wilbert Pumacay, Ranjay Krishna, Adithyavairavan Murali, Arsalan Mousavian, and Dieter Fox. Robopoint: A vision-

language model for spatial affordance prediction for robotics. In *Conference on Robot Learning (CoRL)*, 2024.

[49] Rowan Zellers, Yonatan Bisk, Roy Schwartz, and Yejin Choi. SWAG: A Large-Scale Adversarial Dataset for Grounded Commonsense Inference, August 2018. URL http://arxiv.org/abs/1808.05326. arXiv:1808.05326 [cs].

[50] Rowan Zellers, Ari Holtzman, Yonatan Bisk, Ali Farhadi, and Yejin Choi. HellaSwag: Can a Machine Really Finish Your Sentence?, May 2019. URL http://arxiv.org/abs/1905.07830. arXiv:1905.07830 [cs].

[51] Shiduo Zhang, Zhe Xu, Peiju Liu, Xiaopeng Yu, Yuan Li, Qinghui Gao, Zhaoye Fei, Zhangyue Yin, Zuxuan Wu, Yu-Gang Jiang, and Xipeng Qiu. VLABench: A Large-Scale Benchmark for Language-Conditioned Robotics Manipulation with Long-Horizon Reasoning Tasks, December 2024. URL http://arxiv.org/abs/2412.18194. arXiv:2412.18194 [cs].

[52] 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, October 2025. URL http://arxiv.org/abs/2510.03827. arXiv:2510.03827 [cs].

TABLE IV: Maximum episode horizon per task. Episodes that reach the horizon without satisfying the success condition are marked as failures. All evaluated policies are run with a policy Δt of 66 ms.

Task	Horizon (Policy Steps)
Pick with Occlusion	300
Pick with Covering	300
Pick with Containment	500
Pack with Occlusion	1500
Pack with Covering	1500
Pick with Affordance	300
Stack with Affordance	600
Pick with Object Permanence	300

VI. Appendix

Below, we include additional details on per-task implementation and sampling, policy parameters for evaluation, and VLM usage.

A. Per-Task Implementation Details

We elaborate on simulation implementation details per commonsense taxonomy task here. First, in Table IV, we share the horizon used per task. Per-task, we also describe environment sampling and task implementation (including details like success condition, etc.).

Pick with Occlusion, Pick with Covering

These environments use MSProc scenes. Scenes are updated with a procedural cluttering step. In this step, all graspable objects within a filtered superset are moved to within a radius of the target object, placed such that no contact exists at the time of placement (via MuJoCo contact checks), and settled.

Between occlusion and covering tasks, the only difference is the parameters used for procedural cluttering. Clutter object placement uses a spherical coordinate system with the task's target pick object as its origin. The primary parameter varied for covering is the polar angle ϕ- a lower value is used for covering, denoting higher z values (and closer to the target object, given a fixed radius r); this achieves clutter object positions on top of and covering the target pick object. In contrast, higher ϕ in the occlusion setting typically achieves clutter objects surrounding but not covering the target pick object.

For both tasks, the success condition is simply a successful lift of the target object (via object z coordinate), combined with a contact check between the gripper and the object.

Pick with Containment

These environments use MSCrafted scenes. Scenes are sampled with a target drawer (articulated object) as well as a target pick object. A position is sampled for the target pick object inside the closed drawer, using the drawer handle as a reference point.

The task success condition has 2 stages. The first stage is the successful opening of the drawer (measuring the degree of articulation of the sliding joint). The second stage is the successful pick of the object (see above for the pick task success condition).

Pack with Occlusion, Pack with Covering

These environments use MSProc scenes. The scene initially includes a single target object, along with an added box receptacle for placement. Scenes then use the same procedural cluttering additional step as above (i.e., same occlusion/covering split) to introduce more target objects for packing. Robot placement during task sampling is done such that both the receptacle and pick objects lie within its workspace.

The success condition of packing tasks is the placement of the newly placed clutter objects and the originally present object into the box. For each object, we do a cascading set of checks to verify box containment. The first check is a contact check with the box, which additionally requires the contact normal to be facing up (dot product threshold with positive z axis in the box's coordinate frame). This check may fail due to either improperly registered MuJoCo contacts or items being stacked in the box. Therefore, we do an additional geometric check, where rays are cast outward from the object origin along the box $\pm x$ and $\pm y$ axes. If all rays eventually intersect with the box geometry, we consider the object as packed. The packing task must have all objects satisfy the packed condition to be marked as successful.

Pick with Affordance

These environments use MSCrafted scenes, with no additional modifications. The key implementation detail with this task lies in its success condition. We leverage the already generated antipodal grasps per object within the MolmoSpaces ecosystem, and annotate the grasps per object for correctness given the task of picking the object by the handle (or picking up the utensil to give to someone, picking up the hot pan, etc.).

Within the task success condition, all grasps for the object are loaded. If the object is successfully lifted and the gripper has made contact, a nearest neighbors check is invoked against the set of grasps. Our distance metric is simply the translation-only distance between the fingertip-midpoint tool center point and a generated grasp's translation component. We consider the 5 nearest grasps, and if ≥ 3 of those lie within the set of grasps classified as "good" given the task, success is returned. Fig. 6 shows an example of the grasp annotations for the task "pick up the hot pan". The nearest neighbors to the current grasp must be those in green (i.e., on the pan's handle) for the task to be a success.

Stack with Affordance

Environments use MSCrafted scenes. We additionally place 2-4 blocks within the robot's workspace in the scene. All blocks are sized differently, with size options including 2 cm, 4 cm, 6 cm, and 8 cm side lengths.

The success condition requires that blocks are stacked in order of size (with the largest on the bottom and the smallest on top). Note that block sizes are sufficiently different such that this is generally required for any stacking success. Concretely, for every possible pair of larger block and smaller block, we verify by both a contact check and a fallback geometric check if the smaller block is immediately above the larger block. If this holds for $n_{\text{blocks}} - 1$ pairs, then our

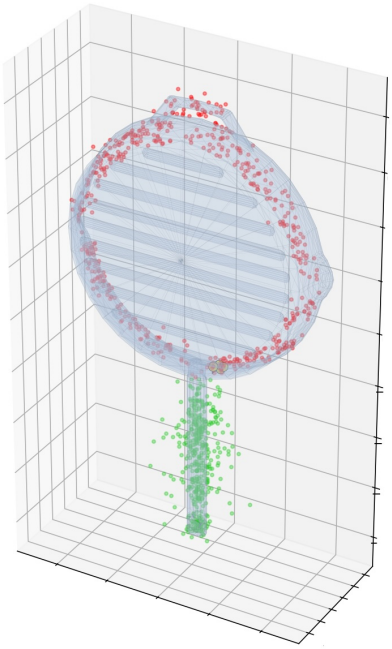

Fig. 6: Grasps classified as good and bad using our grasp annotation tool, given the task "pick up the hot pan." A semantically correct grasp given this command requires picking up the pan by the handle.

success condition returns True.

Pick with Object Permanence

Environments use MSProc scenes. We additionally place 2 upside-down mugs in midair above a surface in the scene, with a ball on the surface immediately underneath one of the mugs. At the first task frame, the mugs are still suspended in midair, which allows policies a window of observing the ball prior to the scene settling and the mugs falling to cover the ball. The initial visibility of the ball is verified by a segmentation mask check on the ball in the first frame of the task.

For the first 5 seconds of simulation time, policy outputs are set to no-operation. In other words, the policy is forced not to act prior to the scene settling and the mugs falling to obscure the ball. The policy still receives observations during this time (necessary for solving the task). From here, the task's success condition is a simple pick task success condition (see above), with the target object being the mug covering the ball.

B. Planner for Task Feasibility Verification

In our task sampling process, after sampling environments, we run a planner-based policy on each. We filter to environments where the planner policy succeeds in order to arrive at a candidate set of environments, from which the final 500-1000 environments per task are chosen. Note that the 100% planner success rate on our tasks in Table II follows by construction from this procedure.

Our planner policy follows a procedure similar to MolmoBot's data generation (for manipulation tasks with DROID). We use the already generated grasps for the target object, sample one of the grasps, and then generate pregrasp, grasp, preplace, and place poses as needed. We then use inverse kinematics to execute this sequence of gripper poses (i.e., no collision avoidance, etc.).

In clutter tasks, this ends up being a sub-optimal way to solve tasks; the gripper may collide with and knock over objects on its way to the target object. However, it at least verifies that the task is solvable. In packing tasks, we attempt to minimize the planner's suboptimality by sequencing the multiple pick and places such that the planner picks clutter objects in reverse order of their placement during cluttering. In a covering setting, for example, the planner would pack the top item in a pile first, before packing the items below.

For the Pick with Containment task, our planner has 2 stages. The first stage generates poses for grasping and pulling the drawer handle, and the second stage generates poses for picking up the object within the drawer.

C. Policy Parameters

Table V shows details on key parameters used per policy in evals. Note that all learned policies tested are joint position policies finetuned on DROID; we use publicly released DROID checkpoints for all. We additionally binarize the gripper output, thresholding at the midpoint of the range of gripper values. We execute the first k actions of each predicted chunk (where k is the per-policy value in Table V) before re-querying the policy. For all learned policies, we terminate the task upon success. For TiPToP, this is unnecessary since the planner executes an open loop before stopping. For more details on all relevant parameters, refer to the table above.

D. More Details on VLM Evaluations

The VLM-judge ground truth consists of Pick with Occlusion policy rollouts annotated by the authors: we sampled approximately 200 episodes per policy, yielding final counts of 204 ($\pi_{0.5}$), 206 (DreamZero), and 201 (MolmoAct2), for $n = 611$ total. All accuracies in Tables III and VI and the confusion matrices in Tables VII and VIII are computed over this set. This annotated set also underlies the failure-mode analysis of Sec. IV-A, where we restrict to the $n = 201$ environments shared across all three policies.

We now elaborate on the classifications of failure modes (or success) used for these evaluations:

- **No Grasp Attempt** - the gripper does not come into contact with any graspable object in the scene
- **Wrong Object** - the gripper comes into contact and/or successfully grasps a different object from the target
- **Success** - the correct object is successfully picked up
- **Clutter Collision** - the robot attempts the correct object but fails due to a collision with a clutter object

Parameter	**MolmoAct2**	π_0**,** π_0**-FAST,** $\pi_{0.5}$	**DreamZero**	**TiPToP**
Policy Type	Learned	Learned	Learned	TAMP (Planner)
Finetuning	DROID	DROID	DROID	N/A
Output Format	Joint Position	Joint Position	Joint Position	Joint Position
Policy Δt	66 ms / 15.15 Hz	66 ms / 15.15 Hz	66 ms / 15.15 Hz	66 ms / 15.15 Hz
Actions Executed per Chunk	8	8	24	N/A, Open-Loop
Cameras	Wrist + Exo	Wrist + Exo	Wrist + Exo	Wrist + Exo, + Depth
Input Image Resolution	640×480 (Full Render)	224×224 (Resize w/ Pad)	180×320 (Resize w/ Pad)	640×480 (Full Render)
Gripper Decode	Binary @ 0.5	Binary @ 0.5	Binary @ 0.5	Binary @ 0.5
End on Success	True	True	True	False

TABLE V: Per-policy evaluation configurations.

TABLE VI: VLM-as-judge fine-grained accuracy (%) across rollouts from three VLA policies. Best per column in **bold**. Models without bibliography entries: GPT-5.4 via the OpenAI API (`gpt-5.4`, accessed 2026-05-28); Qwen3-VL-8B (https://huggingface.co/Qwen/Qwen3-VL-8B-Instruct); Qwen3.6-35B-A3B (https://huggingface.co/Qwen/Qwen3.6-35B-A3B).

Evaluator	$\pi_{0.5}$	DreamZero	MolmoAct2	Mean
Gemini 3.1 (high)	45.1	52.9	45.8	47.9
Gemini 3.1 (low)	46.6	**63.1**	43.8	51.2
Gemini 3.1 (medium)	44.6	53.9	42.8	47.1
Gemini Robotics-ER 1.6 (high)	**51.0**	50.5	52.2	51.2
Gemini Robotics-ER 1.6 (low)	50.5	51.0	**53.2**	**51.6**
GPT-5.4 (high)	45.1	46.1	46.8	46.0
Qwen3-VL-8B	26.5	30.6	47.3	34.8
Qwen3.5-35B-A3B (reasoning)	38.9	44.1	41.1	41.4
Qwen3.6-35B-A3B (reasoning)	42.0	46.5	34.8	41.1
Qwen3.5-122B-A10B	44.6	46.1	44.8	45.2
Qwen3.5-397B-A17B (fp8)	41.1	48.1	36.8	42.0
Molmo2	27.9	37.4	19.4	28.2

- **Grasping Failure** - the robot attempts the correct object but fails to grasp it after making contact

Table VI expands Table III with all evaluated configurations; Table III reports the best configuration per model family by mean accuracy (Gemini 3.1 low, Gemini Robotics-ER 1.6 low, Qwen3.5-122B-A10B). Interestingly, setting Gemini models' thinking budget to higher values did not improve performance.

We also share confusion matrices that elaborate on the points mentioned in Section IV-D. In Table VII, note that in rollout videos where the policy does not attempt to grasp anything (i.e. makes contact with no graspable object in the scene), more than 50% of the time the model outputs that an object has been attempted and a different failure mode occurs. Table VIII shows that Gemini ER 1.6 misclassifies successful trajectories as grasping failures a majority of the time. For more details on the various failure modes, refer to the confusion matrices below.

Below, we share the prompt used for all VLM evals. Each query consists of 9 images: a goal image (an early frame with the target object boxed in red, plus a zoomed-in crop) followed by 8 frames sampled uniformly from the rollout. We parse the predicted class from the VLM inference output. The prompt's output labels map to our categories as: Grasping success → Success; Grasping failure, correct object → Grasping Failure; Grasping failure, clutter → Clutter Collision; Grasping failure, wrong object → Wrong Object; No grasp attempt → No Grasp Attempt.

TABLE VII: Aggregated confusion matrix for Gemini 3.1 (low reasoning) judged across all rollout sources (DreamZero, $\pi_{0.5}$, MolmoAct2; n=611). Rows are ground-truth labels, columns are model predictions; diagonal entries (correct predictions) are in **bold**.

True \ Pred	No Grasp Attempt	Wrong Object	Success Success	Clutter Collision	Grasping Failure	Total
No Grasp Attempt	**87**	59	22	2	7	177
Wrong Object	15	**126**	10	3	3	157
Success	1	8	**77**	1	5	92
Clutter Collision	0	15	10	**1**	0	26
Grasping Failure	11	49	74	3	**22**	159
Total	114	257	193	10	37	611

TABLE VIII: Aggregated confusion matrix for Gemini Robotics-ER 1.6 (low reasoning) judged across all rollout sources (DreamZero, $\pi_{0.5}$, MolmoAct2; n=611). Rows are ground-truth labels, columns are model predictions; diagonal entries (correct predictions) are in **bold**.

True \ Pred	No Grasp Attempt	Wrong Object	Success Success	Clutter Collision	Grasping Failure	Total
No Grasp Attempt	**98**	43	0	1	35	177
Wrong Object	31	**104**	1	1	20	157
Success	7	9	**12**	2	62	92
Clutter Collision	2	10	0	**3**	11	26
Grasping Failure	23	25	8	5	**98**	159
Total	161	191	21	12	226	611

VLM-as-judge Evaluation Prompt:

```
I want you to evaluate the performance of a vision language action
model on the given robotics task (a cluttered scene task in a
MuJoCo-based simulation benchmark). The task of the robot is to pick
up a particular object within the cluttered scene. I will pass in 9
images relevant to the task and the robot's policy rollout. The first
image depicts the target object that needs to be picked up. It
includes a frame early in the task, with the target object boxed in
red. That image also includes a zoomed in crop of the target object.
The next 8 images shown are in chronological order and come from the
robot executing the task. Carefully analyze these sequential images
and classify the success or failure mode that the policy experiences-
i.e. whether the robot is successful in picking up the target object
specified in the first image, based on how the rollout progresses in
the last 8 images. Use the following partition of the outcome space
for this task:

The robot doesn't attempt to grasp anything- the black robot gripper
does not make contact with any object in the scene, only empty
surfaces or nothing at all

Attempts to grasp an object in the scene ->
a. Does it go for the wrong object- this requires the black robot
   gripper making contact with an object OTHER THAN THE TARGET OBJECT
   specified in the first image of the goal object boxed, but does
   not require a good grasp
b. Does it go for the right object- this requires the black robot
   gripper making contact with the target object specified in the
   first goal image.
   i.  Does it successfully grasp the object- this requires the
       gripper closing around the object
   ii. Does it fail- the gripper does not successfully close around
       the object, but it still makes contact with the object
       1) Does it fail due to clutter- As the black robot gripper
          moves toward the target object, does it run into another
          object along the way?
       2) Does it just have a grasping failure, where the gripper
          approaches and makes contact with the correct object, but
          fails to close around the object

In addition to your reasoning, you should provide a final
classification of this episode as one of the below, given the above
partition of the outcome space:

Grasping success
Grasping failure, correct object
Grasping failure, clutter
Grasping failure, wrong object
No grasp attempt

Do your diagnosis after carefully considering these factors:
- Always be aware of which object is the target. For this, refer to
  the first image passed in, and the object boxed in red and with a
  zoomed in crop.
- The path the policy takes. Does it run into other objects on its
  path toward the target object
- Does the robot stop and attempt to grasp other objects, or does it
  keep going towards the target object
- Closely examine the black robot gripper in all frames. What is it
  holding, if it is holding anything? Does the held item match the
  target item boxed in red on the right side of each image? Is it
  fully closed around the object? What is the gripper making contact
  with, if it is making contact with anything?
```

Fig. 7: VLM-as-judge evaluation prompt used across all VLM judge experiments.