GROUNDPROBE: A BENCHMARK FOR DIAGNOSING LANGUAGE GROUNDING IN ROBOTIC POLICIES

Anonymous authors
Paper under double-blind review

ABSTRACT

Vision-Language-Action (VLA) models report high success rates on standard manipulation benchmarks, yet recent stress tests reveal that this success often collapses under modest perturbations of objects, viewpoints, or language, suggesting that headline numbers conflate trajectory memorization with functional understanding. We introduce GroundProbe, a diagnostic benchmark that decomposes policy evaluation into three complementary dimensions, namely (1) language grounding, (2) compositional generalization, and (3) state-conditioned behavior. Within each dimension, training and testing splits are constructed so that the building blocks for the testing distribution are present in training, and each testing condition is matched to a seen one that differs in a single controlled factor, so that, given competence on the matched seen condition, a drop on the held-out one is evidence of a failure to transfer rather than of insufficient supervision. GroundProbe is built on NVIDIA Isaac Lab-Arena with a Franka Panda manipulator and a curated set of $\sim$30 meshes, totaling roughly 2,500 training trajectories and 1,770 testing episodes spread across six sub-tasks. Rather than reducing a policy to one number, GroundProbe reports closed-loop success rates together with normalized transfer gaps, a gap being the fall from a seen composition to the held-out one that differs from it in a single controlled factor, which distinguishes failure modes such as direction being entangled with category from the absence of an abstract category representation. The release comprises the scenes, instruction sets, teleoperated demonstrations, evaluation configurations and the analysis code, giving VLA designers a sharper diagnostic instrument than overall task success rate. We report a first datapoint rather than a full diagnostic profile. Populating the three axes across a slate of current models is the work this release is meant to enable. The benchmark, the demonstrations and the analysis code will be released upon acceptance.

1 INTRODUCTION

Vision-Language-Action (VLA) models have advanced rapidly. The OpenVLA series (Kim et al., 2024; 2025), the π_0 family (Black et al., 2025; Physical Intelligence et al., 2025), GR00T N1 (Bjorck et al., 2025), SmolVLA (Shukor et al., 2025), Gemini Robotics (Gemini Robotics Team, 2025), and VLA-0 (Goyal et al., 2025) have reported strong success rates on standard manipulation benchmarks. Recent stress tests, however, expose substantial brittleness: variations in camera viewpoint, robot initial state, or distractor placement can reduce success rates from above 95% to below 30% (Fei et al., 2026), while additional instruction rewrites can drive them to 0% (Zhou et al., 2025). These results raise a question about language grounding. A policy may perform well with familiar instructions yet fail when the same goal is expressed differently, suggesting that it relies partly on contextual cues or learned action patterns rather than a robust, compositional understanding of the request.

An aggregate success rate cannot reveal whether a model treats "left" and "right" as spatial relations that transfer across objects and scenes, applies a learned skill to a previously unseen object pair, or chooses a skill based on the scene's physical state. Existing benchmarks often vary these factors together, making failures difficult to interpret. A diagnostic benchmark instead needs matched conditions that differ in one controlled factor. We introduce GroundProbe, which evaluates three complementary dimensions under a common protocol. Each testing condition uses objects, relations, and sub-skills observed during training, but tests a composition withheld from training. We compare the same policy on that composition and a matched, seen composition. 'D1' tests language ground-

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

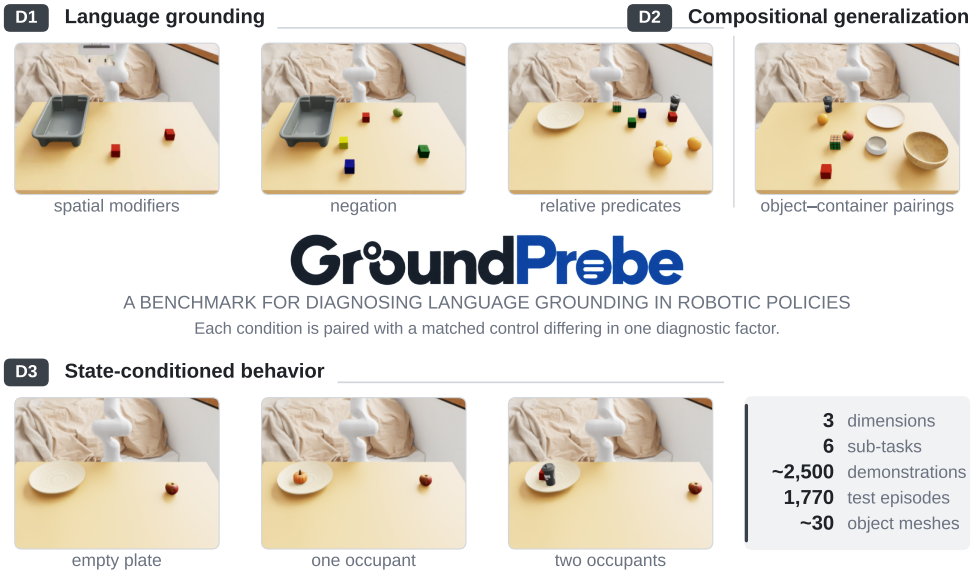


Figure 1. The conditions GroundProbe evaluates. Each cell shows a scene from one of the three dimensions together with the construct it probes, and every condition withholds a composition whose constituents all appear in training under other combinations.

ing through attribute composition, spatial modifiers, negation, and relative predicates. 'D2' tests compositional generalization across object–container pairings. 'D3' tests state-conditioned behavior: a fixed instruction requires different sub-skill sequences depending on the scene state. Together, these dimensions distinguish three sources of failure that aggregate success rates conflate: linguistic binding, compositional recombination, and state-dependent skill selection.

GroundProbe is implemented in NVIDIA Isaac Lab-Arena (NVIDIA Isaac Lab-Arena Contributors, 2025) with PhysX contact dynamics, a shared Franka Panda embodiment, and a curated library of approximately 30 object meshes. It is intended as a diagnostic benchmark rather than a leaderboard. We report closed-loop success rates separately for each diagnostic condition; the primary evidence is the performance gap between matched conditions that differ in one controlled factor, such as a composition seen during training versus one withheld from it.

Our contributions are:

- A contrastive evaluation protocol designed to distinguish failures of language grounding, compositional recombination, and state-dependent skill selection through matched-condition performance gaps.
- GroundProbe, a reproducible Isaac Lab-Arena implementation that includes environments, instruction sets, success criteria, approximately 2,500 teleoperated demonstrations with synchronized RGB observations and proprioceptive data, 1,770 test episodes across six sub-tasks, and an analysis pipeline for computing condition-level results.
- Case studies involving a language-free policy, two released VLAs evaluated zero-shot, and a fine-tuned VLA. These comparisons help distinguish failures in basic workspace control from failures to generalize to withheld compositions.

2 Related Work

Benchmarks that measure task success. Manipulation benchmarks have developed along two broad lines. The first emphasizes skill diversity, including Meta-World (Yu et al., 2020), RL-Bench (James et al., 2020), Push-T from Diffusion Policy (Chi et al., 2023) and implicit behavioral cloning (Florence et al., 2021), CALVIN (Mees et al., 2022), and LIBERO (Liu et al., 2023). The second broadens task coverage for evaluating generalist policies, including SimplerEnv (Li et al., 2024), RoboCasa (Nasiriany et al., 2024), RoboTwin and RoboTwin 2.0 (Mu et al., 2025; Chen et al.,

2026b), VLABench (Zhang et al., 2025), and RoboDojo (Chen et al., 2026a). ARNOLD (Gong et al., 2023) and COLOSSEUM (Pumacay et al., 2024) additionally measure performance under scene and object perturbations. These benchmarks identify conditions under which success declines, but task success alone does not establish which cues a policy uses. Moreover, broad task coverage makes it difficult to pair each testing condition with a training condition that establishes which of its components were learned. VLMbench (Zheng et al., 2022) is particularly close in intent: it evaluates language-guided manipulation across compositional variations in color, size, relative position, and direction. GroundProbe builds on these factors but designs the training split and testing conditions jointly. Each held-out composition contains constituents observed during training and is compared with a matched seen composition. By restricting the action vocabulary and scene family, this design makes the resulting performance gap more directly informative about the construct under test.

Diagnostic and contrastive evaluation. Recent benchmarks examine failure modes beyond aggregate success. RoboEval (Wang et al., 2025) reports efficiency, coordination, stability, and stagewise progress, helping locate execution failures without directly testing which parts of an instruction govern behavior. LIBERO-Plus (Fei et al., 2026) and LIBERO-PRO (Zhou et al., 2025) measure robustness under controlled perturbations of evaluation conditions. ATOM-Bench (Wu et al., 2026) is especially relevant because it distinguishes execution from instruction understanding through atomic skills and held-out compositions; its real-robot evaluation, however, is subject to variation across physical setups. RoboHiMan (Chen et al., 2025) distinguishes planner-level from policy-level failures in long-horizon tasks, while ClevrSkills (Haresh et al., 2024) evaluates compositions of skill graphs across a broad task hierarchy. One Word, Different Action (Liu et al., 2026) uses matched instructions to test whether a linguistic change produces the required change in action on real hardware. GroundProbe applies a related contrastive principle to training-time composition: it withholds specific combinations while retaining their constituents in training, then compares each held-out condition with a matched seen condition. These combinations include direction–category pairings, negated referents, relative predicates applied to held-out object sets, and object–container pairings. The approach follows the logic of controlled vision-language probes, in which a minimal contrast supports a more specific interpretation than aggregating accuracy alone (Thrush et al., 2022; Yuksekgonul et al., 2023). A table in the supplementary material compares benchmark properties directly.

Simulation infrastructure. GroundProbe is implemented in NVIDIA Isaac Lab-Arena (NVIDIA Isaac Lab-Arena Contributors, 2025), built on Isaac Lab (Mittal et al., 2023). Its separation of scene, embodiment, and task supports a common evaluation protocol across six sub-tasks. The supplementary material explains the choice of simulation stack.

3 The GroundProbe Benchmark

Table 1 states the whole benchmark in one place: what each dimension evaluates, which composition is withheld from training, and the pair of conditions its metric compares. Its last two columns count trajectories for training and episodes for testing. The subsections that follow expand each row.

3.1 Diagnostic Principle

Three design principles apply across all three dimensions. First, every testing condition is compared with a matched condition that differs in one controlled factor, while the relevant constituents remain present in training. Most conditions test a withheld composition of instruction components or, in D2, scene components. Attribute composition is an exception: it compares a full instruction with single-attribute probes in the same scene to test whether both attributes influence the policy's behavior. Second, paired conditions share the same objects, layout, execution requirements, and success criterion, so the targeted composition is the intended difference. Third, all dimensions use a small, shared vocabulary of primitive sub-skills to limit variation in low-level manipulation demands. D3 deliberately varies the number of primitives required, because selecting the appropriate sequence is itself the capability under test. The supplementary material details these principles and their rationale.

Table 1. GroundProbe at a glance. Each dimension names what it evaluates, which composition training withholds, and the pair of conditions the metric compares.

Dim.	Evaluation scope	Held-out composition	Metric	Train	Test
D1	language grounding over attributes, spatial modifiers, negation and relative predicates	the composition alone, a direction and category pair, a negated referent or a relative predicate applied to a held-out object set, with attributes contrasted against a single-attribute probe	seen against held-out composition, one contrast per construct	1,850	1,170
D2	the binding of a source object to a container	5 of the 15 cells of a 5×3 matrix, one per source, each source keeping two trained containers	seen against held-out pairing	500	300
D3	skill selection under a fixed instruction and a varying scene state	the occupant identity in V2 and V3, so occupancy cannot be read off a memorized object	seen against held-out occupant	150	300

3.2 Evaluation Protocol

A testing episode specifies an environment configuration, not a stored trajectory. The scene is instantiated from a seed, the instruction is issued, and the policy runs in closed loop until it meets the success criterion or reaches the step limit. Thus, the testing counts reported in the supplementary material denote rollouts to be executed. Per-step action error against a reference trajectory is less suitable for this diagnosis: a policy may follow much of the reference trajectory yet act on the wrong object.

GroundProbe reports closed-loop success rates and normalized transfer gaps without combining them into a single score. Policies with similar mean success rates can have different failure profiles. We use the same gap wherever a matched pair is defined, while retaining the condition-level results needed to interpret each dimension. D1 reports contrasts for individual linguistic constructs; D3 reports success separately for each scene state and compares outcomes across occupant states.

For each dimension, evaluation runs matched conditions on the same layouts. The conditions differ in the factor under test: for example, an instruction term in D1, an object–container pairing in D2, or the plate's occupant in D3. An episode succeeds when, after release, the addressed object's center remains within a calibrated radius of the container's current position for 30 consecutive control steps. The same criterion is used to accept demonstrations and score policy rollouts. Layouts that place the target inside its destination container at initialization are rejected before rollout, because they could satisfy the criterion without any policy action (the supplementary material).

Let SR_A denote the closed-loop success rate under condition A. For a matched pair, the normalized transfer gap is $G = 1 - \frac{SR_{\text{held-out}}}{SR_{\text{seen}}}$. Rates are compared within the same complexity tier, and each gap is reported alongside its two constituent rates. If $SR_{\text{seen}} = 0$, the gap is undefined. A condition is marked "not measured" only when all sampled layouts are rejected before rollout; it is never assigned a success rate of zero. When a dimension has multiple contrasts, they are interpreted jointly. For example, a smaller direction gap than category gap in D1 suggests stronger transfer of the spatial term than of the category term, conditional on adequate success in the matched seen conditions.

Interpreting a transfer gap requires evidence that the policy can act in the workspace. We therefore evaluate policies in three stages: Stage 0 checks reaching and grasping with the shared embodiment; Stage 1 measures success on the matched seen condition; and Stage 2 evaluates the transfer contrast. If a policy has no successes on the seen condition, its gap is undefined, as occurs for both zero-shot policies in Section 4. Reporting the stage at which evaluation breaks down helps distinguish a workspace mismatch from a failure of transfer. Because seen-condition competence is continuous rather than binary, we always report both underlying rates and flag gaps based on fewer than five seen-condition successes as weakly supported.

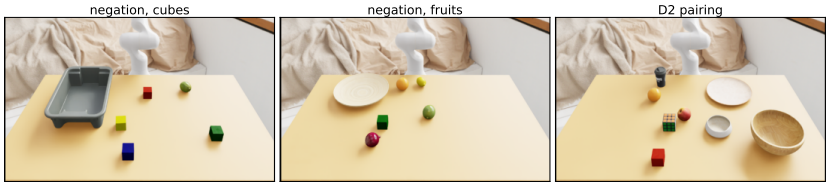


Figure 2. The two mirrored negation scenes and the static D2 scene. Whichever category supplies four instances is the one the negation scopes over, and D2 keeps every source and container on the table so that only the instruction identifies the target.

3.3 Language Grounding (D1)

D1 evaluates four distinct language constructs; that is, attribute composition, spatial modifiers, negation, and relative predicates, rather than treating them as an ordered scale. Each construct is evaluated through matched conditions, but the controlled contrast differs by construct, as the supplementary material sets out. Reporting them separately distinguishes failure modes that a single score would conflate, such as ignoring negation and associating a relative size predicate with a particular object mesh.

Attribute composition. Each scene contains eight objects spanning four categories and two colors. All eight color–category pairings appear in the demonstrations, and each attribute occurs in multiple combinations. To test whether both attributes influence behavior, we compare the full instruction with probes that name only the color or only the category in the same scene. The probes expose policies that respond to one attribute but do not reliably bind the two.

Spatial modifiers. This construct tests whether *left* and *right* transfer as spatial relations across object categories or function as category-specific associations. Each scene contains two instances of each of five categories. During training, each category is paired with one direction: left for the red and green blocks and the Rubik's cube, and right for the blue and yellow blocks. Object positions are reshuffled across demonstrations, so a direction does not identify a fixed location. This design is repeated at three scene complexity levels. Every category and both directions appear in training; the five category–direction combinations using the opposite direction are withheld. We report two complementary contrasts: one changes direction while holding category fixed, and the other changes category while holding direction fixed. Section 3.2 describes how these gaps are interpreted together.

Negation. This construct tests whether a policy can use negation to identify a target rather than relying on its visual salience. Each scene contains four instances of one category and a single object from another category, making the negated referent unique. Two mirrored scene configurations (Figure 2, left and center) vary which category occupies each role. Each training demonstration is labeled with both a direct instruction and a negated instruction referring to the same target. The held-out condition assigns negation to a minority-object identity that was targeted by direct naming, but never by negation, during training. Because the intended target is also the visual odd-one-out, dummy-language and truncated-instruction controls test whether the policy follows the operator or selects the odd object regardless of the instruction (the supplementary material).

Relative predicates. This construct tests whether size predicates are evaluated relative to the objects in the current scene. Each scene contains three differently scaled instances of one fruit type, keeping color, category, and mesh identity constant while varying size. Three clutter levels change visual search difficulty without changing the target (Figure 3, top). Training covers three fruit types and all three predicates; a fourth fruit type is withheld from training. We compare predicate performance on the trained types with performance on the held-out type. Errors concentrated on the middle-sized object may indicate reliance on absolute size cues, whereas failures across all three predicates may be consistent with memorized fruit–predicate associations. These patterns are diagnostic hypotheses rather than unique explanations of a policy's behavior.

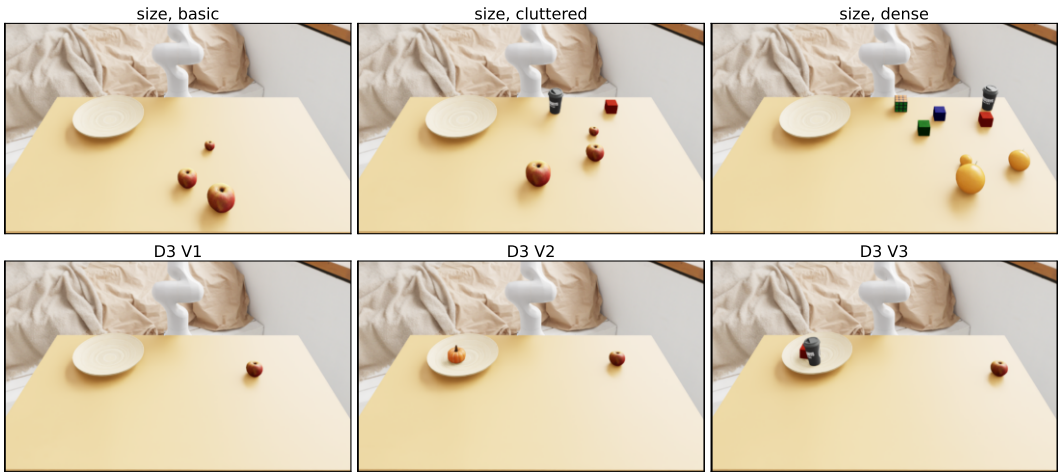


Figure 3. The three relative-predicate clutter tiers, top, and the three D3 states, bottom. Clutter changes the visual search without changing which fruit is largest, and in D3 only plate occupancy varies, which is what sets the required sub-skill.

Table 2. The D2 pairing matrix. Filled circles are trained cells and open circles the one held out in each source row.

Source	plate (on)	bowl (in)	tray (on)
cup	●	●	○
apple	○	●	●
orange	●	○	●
red block	●	●	○
rubik's cube	○	●	●

3.4 Compositional Generalization (D2)

D2 evaluates compositional generalization across a 5×3 matrix of source objects and destination containers (Table 2). Ten object–container pairings appear in training, and five are held out, one for each source object. Thus, every source has been picked, every container has served as a destination, and each held-out pairing can be compared with two trained pairings involving the same source.

The scene composition remains fixed across D2 episodes: all five sources and three containers are present, while their positions are resampled (Figure 2, right). The instruction specifies the destination. Keeping all containers visible prevents the policy from identifying the target merely by detecting which container is present; resampling positions prevents it from using a fixed location. The containers are geometrically distinct, and two share the preposition *on*, so that preposition alone does not identify either one. Four of the five held-out pairings use *on* and have a trained, same-source control with the same preposition, making the object–container pairing the critical change. The fifth pairs the orange with the bowl. Because *in* uniquely identifies the bowl in this scene, we interpret that contrast with this additional cue in mind. The contrast is therefore read within a row of the matrix rather than across policies.

3.5 State-Conditioned Behavior (D3)

D3 evaluates skill selection under a fixed instruction: "Place the apple on the plate." In V1, the plate is empty and the apple can be placed directly. In V2, the policy must first move one occupant; in V3, it must clear two (Figure 3, bottom). The instruction remains constant, while the visible scene state determines the required sequence of sub-skills. D3 therefore tests responses to observable state rather than inference about hidden state.

During training, plate occupants are resampled from a pool of three objects. In the held-out condition, an occupant is replaced with a mesh never seen in that role. We report success separately for V1, V2, and V3 because completing each variant requires the appropriate sequence. Within a variant, we also

compare scenes with trained and held-out occupants. A drop in this matched contrast suggests that skill selection depends on occupant identity, rather than solely on the number of objects to clear.

Statistics. Each training cell contains 50 demonstrations; the totals in the supplementary material follow from the number of cells.

3.6 Environment and Dataset

All reported evaluations use a Franka Emika Panda with a parallel gripper. The embodiment is a replaceable Isaac Lab-Arena component, and the release includes an alternative binding for Enactic's OpenArm. Scenes, instructions, layout constraints, and success criteria are specified in world coordinates rather than in robot-specific terms. The base scene contains a procedural table and objects drawn from a curated library of approximately 30 meshes from YCB (Calli et al., 2015), Objaverse (Deitke et al., 2023), RoboTwin-OD (Chen et al., 2026b), and Blender-authored assets. Each object is annotated with category, color, and nominal size. The supplementary material illustrates the scenes at each complexity level.

Data collection. Demonstrations are collected through human teleoperation. An operator controls the end effector with a Quest 2 controller through Isaac Lab's differential inverse-kinematics controller. Each episode records synchronized third-person and wrist RGB observations, proprioception, and actions in both supported encodings. Teleoperation permits multiple ways to complete a layout and establishes that an operator can satisfy the same success criterion used to score policy rollouts.

4 Experiments

Four policies are evaluated: Action Chunking with Transformers (ACT) (Zhao et al., 2023) in each released action encoding, which takes no language input; pi0.5-DROID and MolmoBot-DROID run zero-shot; and pi0.5-DROID fine-tuned on GroundProbe demonstrations. All of them are rolled out closed-loop under one control block, on one shared set of layouts drawn from outside the collection range, and The supplementary material lists the settings.

4.1 Scripted controls and a policy without language

Before interpreting gaps for learned policies, we test whether the diagnostic contrasts respond to failures specified in advance. Table 4 evaluates four scripted policies, each assigns a fixed behavior before evaluation: an oracle, a policy that ignores the instruction, a policy that binds direction to category, and a policy that binds category to direction. The oracle has no gap; the instruction-ignoring policy has a gap of one on both contrasts; and each binding policy has a gap of one only on the contrast targeting its defect. This check validates the sensitivity of the contrasts to these specified behaviors, rather than establishing their ability to identify every possible failure mechanism. The supplementary material describes the policies.

We then apply the same contrasts to learned policies. For each sub-task and released action encoding, we train an ACT policy using demonstrations from that sub-task alone. ACT receives vision and proprioception but no instruction. The supplementary material reports success for every cell. On the layouts used for demonstrations, the joint-action encoding achieves 72% overall success and the delta-pose encoding achieves 58%. On fresh layouts from the same cells, success falls to 21% and 16%, respectively. The tools cell retains the strongest performance; it has one instruction and the fewest distractors.

Table 3 reports the corresponding contrasts. Three have nonzero seen-condition success rates, and each yields a gap of one: reversing the instructed direction, changing the negated referent to an identity never negated during training, and replacing the plate occupant with an object never seen in that role each reduce success to zero. Changing the category term has little effect in a tier where only the addressed category appears on the table. Other gaps are undefined because their matched seen conditions have zero successes. In D2, where every episode contains the same five sources and three containers and the instruction specifies the pairing, ACT achieves 12% success on trained compositions and 0% on fresh layouts.

Table 3. ACT, the control without language. Each row is one contrast, measured on one shared set of layouts in both released action encodings. A dagger marks a gap whose seen condition carries fewer than five successes.

		joint			delta-pose	
Dim.	What the contrast changes	seen	unseen	Gap	seen / unseen	Gap
D1	the direction term of the instruction	20%	0%	1.00	30% / 0%	1.00
	the category term of the instruction	20%	15%	0.25	30% / 30%	0.00
	the negated referent	17%	0%	1.00†	11% / 0%	1.00†
	the object set of a relative predicate	0%	5%	-	0% / 0%	-
D2	the container the source is paired with	0%	0%	-	0% / 0%	-
D3	the identity of the plate's occupant	11%	0%	1.00†	not measured	

Table 4. Spatial-modifier gaps for policies whose failure mode is fixed in advance. Each score is the value that policy's defect predicts, and the four rows give the four readings the pair of gaps is specified to separate.

Scripted policy	seen	direction transfer	category	$G_{\mathrm{dir}}, G_{\mathrm{cat}}$
oracle	100%	100%	100%	0.00, 0.00
language-blind	100%	0%	0%	1.00, 1.00
direction-entangled	100%	0%	89%	1.00, 0.11
category-entangled	100%	89%	0%	0.11, 1.00

These results illustrate why success must be interpreted alongside scene structure and matched contrasts. When the scene leaves only one plausible target, a policy without language input can still achieve 88–95% success on the relative-predicate cells. Its performance deteriorates when the instruction is needed to distinguish targets. The ACT results thus provide a language-free reference for interpreting the VLA evaluations that follow.

4.2 Released VLAs and evaluated zero-shot

Evaluated zero-shot on GroundProbe, at their respective expected control rates, neither released policy completes an episode. pi0.5-DROID achieves 0 successes in 167 rollouts across the sampled sub-tasks; one additional sub-task is unmeasured because all sampled layouts fail the spawn gate. MolmoBot-DROID is evaluated on the cubes T1 cell at the clean tier: across eight episodes spanning four serving conditions, it exhibits the same behavior and completes none. Given this small sample, the MolmoBot result is treated as a trajectory diagnostic rather than an estimate of its overall success rate.

The trajectories clarify where execution fails. With the camera pose used during its training, pi0.5 approaches the target, descends to the table, and closes its gripper. It comes within 5 cm of the target in 18 of the 144 episodes with recorded target-distance measurements, but never completes a grasp and place. MolmoBot returns to its demonstration initial configuration and remains there. Neither policy reaches Stage 1 competence, so neither yields an interpretable transfer gap. The supplementary material provides the trajectory traces, measurements, and serving details. These results show that adaptation to this evaluation setting is needed before the policies' language-grounding behavior can be assessed through the proposed contrasts.

4.3 A Language-Conditioned Policy

We therefore fine-tune pi0.5-DROID with LoRA, using the same training recipe in one run per dimension. In the spatial-modifier task, the fine-tuned policy achieves 53% success across five trained category–direction pairings. Reversing the direction word for those categories while keeping the scenes, objects, and success criterion fixed reduces success to 2%, yielding a normalized gap of 0.96. Success across the trained pairings shows that the policy can act on multiple instructed referents in this scene; the matched contrast shows that this performance does not transfer to the withheld category–direction pairings.

Table 5. pi0.5-DROID after fine-tuning on GroundProbe demonstrations, one run per dimension. The contrasts and the layouts are those of Table 3, and a dagger again marks a gap resting on fewer than five seen successes.

Dim.	What the contrast changes	seen	unseen	Gap
D1	the direction term of the instruction	53%	2%	0.96
D2	the container the source is paired with	0%	2%	-
D3	the identity of the plate's occupant	22%	22%	$0.00^{\dagger}$

In D2, the same fine-tuning recipe yields 0% success on trained object–container pairings, so the transfer gap is undefined. In D3, success is 22% both with trained plate occupants and when an occupant is replaced by an object unseen in that role. The resulting gap is zero, but it rests on only two successes in nine episodes per condition and provides weak evidence for occupant-invariant behavior. The three fine-tuning runs use the same recipe and differ in their dimension-specific training data.

5 Limitations

The evaluation here covers one language-conditioned policy, one architecture without language and two released VLAs run zero-shot, enough to show the instrument separating them and not enough to fill the three axes. Several cells rest on few successes, the state gap of zero for the fine-tuned policy among them at two successes in nine on each side, which is too little to conclude that its skill selection survives a change of occupant. A diagnostic profile needs a slate of language-conditioned models trained and evaluated on GroundProbe, which is the work this release is meant to enable. Being simulated is what lets any group run the identical evaluation, and the boundary that comes with it is that a score speaks to language conditioning and skill selection rather than to hardware behavior.

6 Conclusion and Future Work

GroundProbe is a three-dimension benchmark whose splits are designed so that, given competence on the matched seen condition, a drop from the seen to the held-out condition is evidence of a construct-specific transfer failure rather than of missing supervision. What it observes is behavior, so a gap identifies a behavioral signature consistent with a representational failure rather than the representation itself. The next step is a slate of current VLAs, which also tests whether the axes move independently, and the wider aim is to move evaluation from asking whether a policy succeeded to asking what its success shows.

AI use statement

In this work we used large language model assistants for two purposes. In the software, we used them only to debug code and for parts of the testing and verification, and we read, ran and tested every such contribution before it entered the release. In the manuscript, we used them to polish wording. We determined and verified the benchmark design, the scientific claims and every reported number ourselves, and we take responsibility for the final content of this work.

Ethics statement

GroundProbe is a simulation benchmark. The demonstrations were recorded by the authors through teleoperation and contain only simulated robot states and rendered images, and no personal data or human-subject recordings are included. We foresee no direct harmful application beyond the general dual-use considerations of robot learning research.

Reproducibility statement

The benchmark specification is given in Section 3, and the supplementary material gives the full instruction lists, the object metadata, the layout-sampling constraints, the success criterion and the per-step episode schema, and describes the released environments, dataset and analysis pipeline. Layout seeds are stored with every episode, so each recorded scene can be regenerated.

REFERENCES

Johan Bjorck, Fernando Castañeda, Nikita Cherniadev, et al. GR00T N1: An open foundation model for generalist humanoid robots. *arXiv preprint arXiv:2503.14734*, 2025.

Kevin Black, Noah Brown, Danny Driess, Adnan Esmail, Michael Equi, Chelsea Finn, Niccolo Fusai, Lachy Groom, Karol Hausman, Brian Ichter, et al. π_0: A vision-language-action flow model for general robot control. In *Robotics: Science and Systems (RSS)*, 2025. arXiv:2410.24164.

Berk Calli, Arjun Singh, Aaron Walsman, Siddhartha Srinivasa, Pieter Abbeel, and Aaron M. Dollar. The YCB object and model set: Towards common benchmarks for manipulation research. In *International Conference on Advanced Robotics (ICAR)*, 2015.

Tianxing Chen, Yue Chen, Zixuan Li, et al. Robodojo: A unified sim-and-real benchmark for comprehensive evaluation of generalist robot manipulation policies. *arXiv preprint arXiv:2607.04434*, 2026a.

Tianxing Chen, Zanxin Chen, Baijun Chen, Zijian Cai, Yibin Liu, et al. RoboTwin 2.0: A scalable data generator and benchmark with strong domain randomization for robust bimanual robotic manipulation. In *International Conference on Machine Learning (ICML)*, 2026b. arXiv:2506.18088.

Yangtao Chen, Zixuan Chen, Nga Teng Chan, Junting Chen, Junhui Yin, Jieqi Shi, Yang Gao, Yong-Lu Li, and Jing Huo. Robohiman: A hierarchical evaluation paradigm for compositional generalization in long-horizon manipulation. *arXiv preprint arXiv:2510.13149*, 2025.

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

Matt Deitke, Dustin Schwenk, Jordi Salvador, Luca Weihs, Oscar Michel, Eli VanderBilt, Ludwig Schmidt, Kiana Ehsani, Aniruddha Kembhavi, and Ali Farhadi. Objaverse: A universe of annotated 3d objects. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2023.

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: A progressive robustness benchmark for visual-language-action models. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2026. arXiv:2510.13626.

Pete Florence, Corey Lynch, Andy Zeng, Oscar Ramirez, Ayzaan Wahid, Laura Downs, Adrian Wong, Johnny Lee, Igor Mordatch, and Jonathan Tompson. Implicit behavioral cloning. In *Conference on Robot Learning (CoRL)*, 2021.

Gemini Robotics Team. Gemini robotics: Bringing ai into the physical world. *arXiv preprint arXiv:2503.20020*, 2025.

Ran Gong, Jiangyong Huang, Yizhou Zhao, Haoran Geng, Xiaofeng Gao, Qingyang Wu, Wensi Ai, Ziheng Zhou, Demetri Terzopoulos, Song-Chun Zhu, Baoxiong Jia, and Siyuan Huang. ARNOLD: A benchmark for language-grounded task learning with continuous states in realistic 3d scenes. In *IEEE/CVF International Conference on Computer Vision (ICCV)*, 2023.

Ankit Goyal, Hugo Hadfield, Xuning Yang, Valts Blukis, and Fabio Ramos. VLA-0: Building state-of-the-art vlas with zero modification. *arXiv preprint arXiv:2510.13054*, 2025.

Sanjay Haresh, Daniel Dijkman, Apratim Bhattacharyya, and Roland Memisevic. Clevrskills: Compositional language and visual reasoning in robotics. In *Advances in Neural Information Processing Systems (NeurIPS) Datasets and Benchmarks Track*, 2024. arXiv:2411.09052.

Stephen James, Zicong Ma, David Rovick Arrojo, and Andrew J. Davison. RLBench: The robot learning benchmark & learning environment. *IEEE Robotics and Automation Letters*, 2020.

Moo Jin Kim, Karl Pertsch, Siddharth Karamcheti, Ted Xiao, Ashwin Balakrishna, Suraj Nair, Rafael Rafailov, Ethan Foster, Grace Lam, Pannag Sanketi, Quan Vuong, Thomas Kollar, Benjamin Burchfiel, Russ Tedrake, Dorsa Sadigh, Sergey Levine, Percy Liang, and Chelsea Finn. OpenVLA: An open-source vision-language-action model. In *Conference on Robot Learning (CoRL)*, 2024.

Moo Jin Kim, Chelsea Finn, and Percy Liang. Fine-tuning vision-language-action models: Optimizing speed and success. In *Robotics: Science and Systems (RSS)*, 2025.

Xuanlin Li, Kyle Hsu, Jiayuan Gu, Karl Pertsch, Oier Mees, Homer Rich Walke, Chuyuan Fu, Ishikaa Lunawat, Isabel Sieh, Sean Kirmani, et al. Evaluating real-world robot manipulation policies in simulation. In *Conference on Robot Learning (CoRL)*, 2024.

Bo Liu, Yifeng Zhu, Chongkai Gao, Yihao Feng, Qiang Liu, Yuke Zhu, and Peter Stone. LIBERO: Benchmarking knowledge transfer for lifelong robot learning. In *Advances in Neural Information Processing Systems (NeurIPS) Datasets and Benchmarks Track*, 2023.

Yiwei Liu, Luwei Yang, and Shunbo Lei. One word, different action: A real-robot benchmark for language-conditioned embodied reasoning. *arXiv preprint arXiv:2609.05260*, 2026.

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

Mayank Mittal, Calvin Yu, Qinxi Yu, Jingzhou Liu, Nikita Rudin, David Hoeller, Jia Lin Yuan, Ritvik Singh, Yunrong Guo, Hammad Mazhar, Ajay Mandlekar, Buck Babich, Gavriel State, Marco Hutter, and Animesh Garg. Orbit: A unified simulation framework for interactive robot learning environments. *IEEE Robotics and Automation Letters*, 2023.

Yao Mu, Tianxing Chen, Zanxin Chen, Shijia Peng, Zhiqian Lan, Zeyu Gao, Zhixuan Liang, Qiaojun Yu, Yude Zou, Mingkun Xu, Lunkai Lin, Zhiqiang Xie, Mingyu Ding, and Ping Luo. RoboTwin: Dual-arm robot benchmark with generative digital twins. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2025.

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. In *Robotics: Science and Systems (RSS)*, 2024.

NVIDIA Isaac Lab-Arena Contributors. Isaac lab-arena: Composable environment creation and policy evaluation for robotics. `https://github.com/isaac-sim/IsaacLab-Arena`, 2025. Accessed May 2026.

Physical Intelligence et al. $\pi_{0.5}$: A vision-language-action model with open-world generalization. In *Conference on Robot Learning (CoRL)*, 2025. PMLR 305:17–40, arXiv:2504.16054.

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 (RSS)*, 2024.

Mustafa Shukor, Dana Aubakirova, Francesco Capuano, et al. SmolVLA: A vision-language-action model for affordable and efficient robotics. *arXiv preprint arXiv:2506.01844*, 2025.

Tristan Thrush, Ryan Jiang, Max Bartolo, Amanpreet Singh, Adina Williams, Douwe Kiela, and Candace Ross. Winoground: Probing vision and language models for visio-linguistic compositionality. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2022.

Yi Ru Wang et al. Roboeval: Where robotic manipulation meets structured and scalable evaluation. *arXiv preprint arXiv:2507.00435*, 2025.

Zenan Wu et al. Atom-bench: A real-world benchmark for atomic skills and compositional generalization in manipulation policies. *arXiv preprint arXiv:2606.16826*, 2026.

Tianhe Yu, Deirdre Quillen, Zhanpeng He, Ryan Julian, Karol Hausman, Chelsea Finn, and Sergey Levine. Meta-World: A benchmark and evaluation for multi-task and meta reinforcement learning. In *Conference on Robot Learning (CoRL)*, 2020.

Mert Yuksekgonul, Federico Bianchi, Pratyusha Kalluri, Dan Jurafsky, and James Zou. When and why vision-language models behave like bags-of-words, and what to do about it? In *International Conference on Learning Representations (ICLR)*, 2023.

Shiduo Zhang, Zhe Xu, Peiju Liu, Xiaopeng Yu, Qinghui Gao, Yuan Li, 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. In *IEEE/CVF International Conference on Computer Vision (ICCV)*, 2025.

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

Kaizhi Zheng, Xiaotong Chen, Odest Chadwicke Jenkins, and Xin Eric Wang. VLMbench: A compositional benchmark for vision-and-language manipulation. In *Advances in Neural Information Processing Systems (NeurIPS) Datasets and Benchmarks Track*, 2022.

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.