RoboHarness: A Simple Harness outperforms VLA and World Actions Models

Anonymous authors
Paper under double-blind review

Abstract

The rise of large language models (LLMs) has opened a path toward generalist embodied AI, raising the question of how LLMs can act in the physical world. One straightforward approach is training the model to generate actions: collect robot-action data and video demonstration, refine LLM architecture with an action head, and train a vision-language-action (VLA) model at scale. However, we argue that coding agents have the potential to realize generalist embodied manipulation. All they need is a simple yet effective robot harness. We present *RoboHarness*: a robotic harness that gives LLM agents a visual-geometric control panel so that they can directly understand and invoke embodied tasks. With RoboHarness, Qwen3.8-Flash-Next (Qwen Team, 2026b) outperforms state-of-the-art VLA models on BEHAVIOR Challenge 2025 tasks (Li et al., 2024). The implementation will be released as open source.

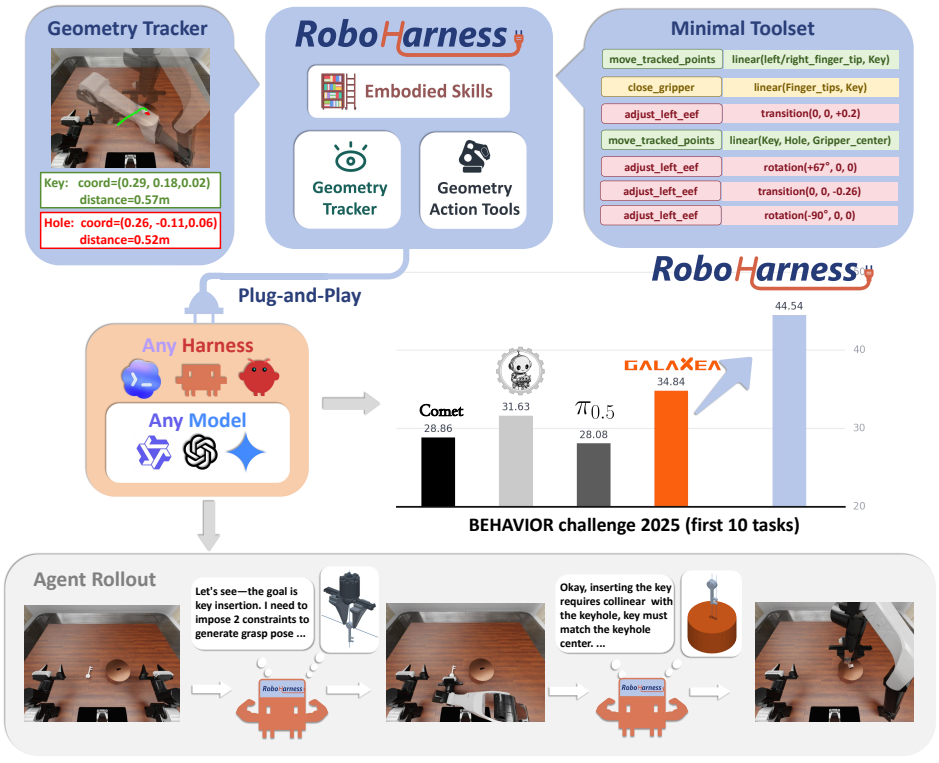

Figure 1: RoboHarness overview: visual keypoints are tracked persistently, converted into geometric constraints, executed, and verified in a closed loop.

1 Introduction

The emergence of large language models (LLMs) opened a path towards generalist robot manipulation. The community has explored many ways to make use of LLM for planning,

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.

visual understanding, physical reasoning, and spatial reasoning to achieve generalizable robot manipulation. Yet an LLM's world knowledge lives in a textual space. A bridge between the LLM and the physical world is therefore required:

One straightforward approach is training the model to generate actions. VLA models try to make the model output end-effector actions directly. Actions may be encoded into the vocabulary for autoregressive generation, as in RT-2, OpenVLA, and FAST (Brohan et al., 2023; Kim et al., 2025b; Pertsch et al., 2025), or grouped into flow-matching chunks, which are later produced by an action head appended to the LLM (Black et al., 2025b; Bjorck et al., 2025). This representation can express arbitrary robot actions, but it requires substantial robot-action data and additional post-training. Such data are difficult to collect and scale to the diversity of real-world objects, scenes, and tasks. This difficulty has motivated world-action models (WAMs), which seek to improve generalization by learning action-conditioned models of how the world changes (Zhao et al., 2026; Wang et al., 2026; Li et al., 2026).

We take a different view. An LLM already has strong capabilities for task decomposition, planning, spatial reasoning, and tool use. The main challenge is therefore not to train the LLM to generate a larger set of actions, but to provide a simple and sufficiently complete interface that connects these capabilities to the physical world.

Existing work provides evidence for this view: Packaging embodied subtasks, such as picking, placing, and navigation, as callable functions allows an LLM to compose them into longer procedures (Liang et al., 2023). ASPIRE shows that an LLM agent can refine such programs and accumulate the repairs in a skill library (Lu et al., 2026). However, the available functions are not complete: their coverage is determined by predefined tasks, objects, and policies. Every new object or task may require another function, leaving the LLM confined to a predefined action space. This suggests that the main limitation lies in the action interface rather than in the LLM's ability.

In this paper, we propose **RoboHarness**. Without training models, RoboHarness combines visual clicks, keypoint tracking, and geometric constraints to provide an embodied control panel for an LLM agent. Concretely, our contributions are:

An expressive, LLM-friendly robot harness that provides an embodied control panel for an LLM agent.

A new paradigm for embodied AI. We show that a generalist LLM, equipped with the right tool harness can outperform VLA and World Action Models

With RoboHarness, the open-source model Qwen3.8-Flash-Next can directly solve long-horizon household tasks in BEHAVIOR-1K (Li et al., 2024) without any finetune, outperforming state-of-the-art models. It also substantially outperforms VLA and WAM baselines on fully out-of-distribution generalization tests.

2 Related Work

Prior work connects pretrained language and vision-language models to manipulation through learned action generation, executable programs, or spatial representations. These approaches differ in where they place the boundary between model reasoning and robot control.

2.1 Vision-Language-Action Models

Autoregressive action tokens. RT-2 and OpenVLA adapt pretrained VLMs to predict discretized robot actions through next-token prediction (Brohan et al., 2023; Kim et al., 2025b). FAST improves this approach by compressing action chunks with a frequency-domain tokenizer, making autoregressive learning more effective for high-frequency control (Pertsch et al., 2025). Galaxea G0.5 further integrates reasoning and action tokens into one autoregressive stream, combining task decomposition, visual grounding, and motor commands within a shared model (Liu et al., 2026).

Continuous action experts. Another family couples a pretrained VLM with a specialized module for generating continuous action sequences. Representative examples include the flow-matching action expert in π_0 and the diffusion transformer in GR00T N1 (Black et al., 2025b; Bjorck et al., 2025). More recent models, such as $\pi_{0.7}$, condition action generation on richer task context, including subtask instructions, subgoal images, and episode metadata, to improve steerability (Physical Intelligence, 2026). These approaches learn a broad motor action space, but require robot-action supervision and additional training. Preserving pretrained knowledge is also a training-design concern: knowledge insulation mitigates interference by blocking action-expert gradients from entering the VLM backbone (Driess et al., 2025). RoboHarness connects an unchanged LLM to robot control through a geometric tool interface, avoiding this additional model-training stage.

2.2 LLMs as Robot Planners and Programmers

Skill and module orchestration. SayCan grounds language-level skill selection in learned affordance values, while Inner Monologue incorporates environmental feedback into replanning (Ahn et al., 2023; Huang et al., 2023b). HarnessVLA combines a frozen VLA for contact-rich manipulation with analytic primitives, using execution memory to guide retries and recovery (Zhang et al., 2026).

Program synthesis and skill discovery. Code as Policies extends composition to generated control programs, including feedback loops and waypoint-level commands (Liang et al., 2023). CaP-X systematically varies the abstraction level of robot APIs and finds that performance degrades as human-designed abstractions are removed; execution feedback and skill discovery help agents work with lower-level interfaces (Fu et al., 2026). ASPIRE compounds program repairs into a growing skill library (Lu et al., 2026).

2.3 Spatial Representations for Manipulation

Visual points and motion proposals. RoboPoint learns to predict image-space affordance points from synthetic supervision (Yuan et al., 2025). PIVOT iteratively selects among visual action proposals with a frozen VLM, and MOKA selects marked keypoints and waypoints (Nasiriany et al., 2024; **?**). HAMSTER predicts a two-dimensional end-effector path and passes it to a separate low-level policy (Li et al., 2025). VoxPoser grounds language in composable three-dimensional value maps and uses model-based planning to generate six-degree-of-freedom end-effector trajectories (Huang et al., 2023a). Constraint-based approaches express the required geometry more explicitly. CoPa grounds object parts in the observed scene and solves geometric constraints over those parts (Huang et al., 2024). OmniManip represents interaction points and directions in an object's canonical space and tracks the resulting six-dimensional poses (Pan et al., 2025). ReKep synthesizes relational constraints over three-dimensional keypoints as Python functions and optimizes subgoals and paths in a closed perception–action loop (Huang et al., 2025). These works establish that grounded geometric relations can support orientation-aware, multi-stage manipulation without task-specific VLM training.

3 Method

Lessons from coding agents show that: a minimal set of simple, orthogonal primitive tools outperforms a collection of predefined features and heavy scaffolding. Inspired by this principle, we argue that an LLM's potential on embodied tasks is best unlocked by giving it the minimal primitive action tools rather than layers of engineered features.

We start from studying spatial relations between a gripper and objects it interacts with, therefore constructing the most elementary primitive action tools. We further prove their completeness in Section 4: (i) **any contact pose** between a gripper and an object can be expressed by geometric constraints, and (ii) once the gripper holds an object, **any spatial relationship** between the in-hand object and a target object can be expressed by geometric constraints.

With the primitive action tools in place, the remaining challenge is spatial understanding: although VLMs cannot reliably reason about 3D spatial relations, their grounding training enables them to localize specific points on objects with reasonable accuracy. We therefore use spatial grounding as the bridge between vision-language models and robots. The LLM marks points of interest on a 2D image and continuously receives their positions via optical-flow tracking and depth back-projection; with this information, it composes accurate primitive actions to perform complex manipulation.

Following this design, we build a simple but effective Robot Harness, which we will release as open source to the community.

3.1 Geometry State for Spatial Grounding

To make primitive action tools more usable to the LLM, we design a persistent geometry-state tracking mechanism that continuously returns the coordinates of objects of interest to the LLM, enhancing its accurate spacial perception capabilities.

3.1.1 Robot coordination definition

We first define a unified camera-frame coordination for diverse robot configurations, making it straightforward for the LLM to read spatial relations:

Axis	Positive direction
x	forward from camera view
y	left from camera view
z	upward from camera view

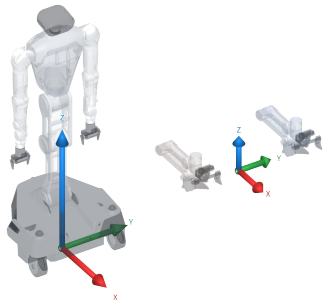

Figure 2: Visualization of the unified coordinate frame on two embodiments: BEHAVIOR R1 Pro (Li et al., 2024) (left) and RoboDojo ARX X5 (Chen et al., 2026) (right).

3.1.2 Track keypoints

With a unified coordinate frame in place, we design the `track_object` tool for LLM to retrieve coordinates of points in which it is interested using no simulation priors the procedure and proceeds as follows:

1. The LLM selects one or more points of interest that are visible in the head camera.
2. Harness continuously tracks each marked point across successive head-camera frames.
3. The latest 2D location is projected onto the depth view to recover the metric distance.
4. The pixel coordinate and depth are then back-projected into the robot camera frame.

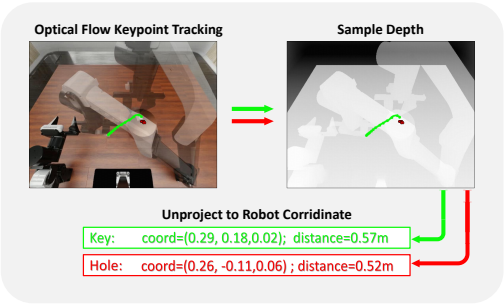

Figure 3: Pipeline of the `track_object` tool: optical-flow keypoint tracking followed by depth back-projection into the robot coordinate frame.

3.2 Action Toolset

Our goal is to design a minimal action toolset.

Therefore, we start from object-interaction relations and build the toolset bottom-up.

A manipulation task can be viewed as a two-stage process: first the gripper establishes a spatial relationship with an object, and then an end-effector action drives this object to satisfy a geometric relationship with another object. For instance, in an insert-key task, the gripper first grasps the key, then moves it until it satisfies the insertion constraint with the lock. Consequently, our action toolset only needs to capture these two stages.

We implement the first stage with keypoint-and-constraint tools: the LLM selects points on the image and specifies the geometric constraints they must satisfy; the harness automatically solves and executes the corresponding actions. To support model fine-tuning, we additionally provide adjustment tools that directly command the robot's low-level degrees of freedom.

In the next section, we will prove that, for any object, any grasp pose can be expressed by two keypoint constraints when the object has not yet been grasped; and when it has already been grasped, assuming the gripper forms a rigid connection with the object, the desired spatial relation between this held object and any other object in the scene can be expressed by two sets of keypoint constraints. Therefore, our action toolset is both minimal and complete.

3.2.1 Keypoint-constraint tools for manipulation

The model supplies a set of input points together with their spatial relations, and the harness automatically solves and executes the corresponding action. The model must distinguish between on-hand and off-hand points: on-hand points lie on the gripper or on a held object and their positions can be modified, whereas off-hand points are fixed in the scene. The harness drives the on-hand points so that they satisfy the spatial constraints specified by the model through the following pipeline:

1. Sample action candidates and select those satisfying the constraints.
2. Discard candidates whose inverse kinematics are infeasible or unreachable.
3. Plan a motion trajectory from the current pose to the target pose.
4. Execute the trajectory.

In principle, the model can express any constraint through coordinate relations among the input points. However, expressions such as collinearity and coplanarity are both verbose to write and frequently required, so we encapsulate the most common patterns as quick requests; their specific modes are listed in Table 1.

Table 1: Modes for keypoint-constraints tool

Mode	**Input points**	**Description**
coordinate	*arbitrary*	move points to satisfy the given coordinate constraints

Quick-requests	**On-hand**	**Off-hand**	**Description**
touch	1	1	an on-hand point coincides with an off-hand point
linear	2	1 or 2	two on-hand points collinear with off-hand points
line-vertical-plane	2	3	an on-hand line is orthogonal to an off-hand plane
plane-parallel	3	3	on-hand plane parallel to off-hand plane
line-vertical-ground	2	0	on-hand line vertical to the ground
face to	3	0	face the target surface
move to reach	0	1	move chassis to make target reachable
grasp	0	1	move chassis to make target reachable

3.2.2 Adjustment tools

To refine robot action or to improve robustness when the constraints fail (for example, when the object slips out of the gripper), we also expose direct low-level robot control interfaces. These interfaces share the same coordinate frame defined in Section 3.1.1, so that the LLM can interpret and use them consistently.

Table 2: Adjustment tools. Displacements are incremental; chassis `spin` is a yaw increment.

Name	Parameters	Meaning
adjust left eef pose	$(\Delta x, \Delta y, \Delta z, r, p, y)$	increment the left end-effector pose or rotation
adjust right eef pose	$(\Delta x, \Delta y, \Delta z, r, p, y)$	increment the right end-effector pose or rotation
adjust chassis	(fwd, side, spin)	move the chassis forward, sideways or spin
adjust height	Δh	raise or lower the body
adjust pitch	$\Delta\theta$	change the body pitch

4 Proof: Completeness of the Keypoint-Constraint Representation

The three points on the robot gripper, namely the left fingertip, right fingertip, and gripper center, are intrinsic points that remain tracked throughout execution (Figure 4). The following results show that constraining their geometric relationships can represent arbitrary motion trajectories and interaction poses. The elementary correspondence between gripper states and keypoint triples is recorded in Appendix **??**.

4.1 Any Grasp Pose Can Be Expressed by Keypoint Constraints

The gripper carries three predefined points $\{F_L, F_R, G\}$: the left fingertip, the right fingertip, and the slider center. A contact pose is (f_l, f_r, θ) with $f_l, f_r \in \partial A$. We prove that (f_l, f_r, θ) is in analytic bijection with three surface points (a_1, a_2, a_3) under the constraints overlap$[a_1, F_L]$, overlap$[a_2, F_R]$, and linear$[c, a_3, G]$, where $c = \frac{a_1+a_2}{2}$. **Claim.** The gripper carries predefined points $\{F_L, F_R, G\}$ and a contact pose is (f_l, f_r, θ) with $f_l, f_r \in \partial A$. We prove an analytic bijection to three surface points (a_1, a_2, a_3) under overlap$[a_1, F_L]$, overlap$[a_2, F_R]$, and linear$[c, a_3, G]$, where $c = \frac{a_1+a_2}{2}$.

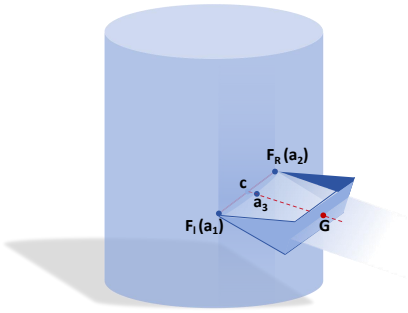

Figure 4: Three predefined gripper points $\{F_L, F_R, G\}$ and a contact pose (f_l, f_r, θ)

(i) $(f_l, f_r, \theta) \mapsto (a_1, a_2, a_3)$. From the two overlap constraints, $a_1 = f_l$ and $a_2 = f_r$. From linear, a_3 satisfies

$$a_3 = c + k\big(p(G) - c\big), \qquad a_3 \in \partial A,\ k > 0. \tag{1}$$

Hence (a_1, a_2, a_3) is uniquely determined.

(ii) $(a_1, a_2, a_3) \mapsto (f_l, f_r, \theta)$. From overlap, $f_l = a_1$ and $f_r = a_2$. Let $u := (a_2 - a_1)/\|a_2 - a_1\|$, fix a right-handed orthonormal basis $(\mathbf{v}_1, \mathbf{v}_2)$ of $u^\perp$, and set $h := \|p(G) - c\|$. Then

$$p(G, \theta) = c + h\big(\mathbf{v}_1 \cos\theta + \mathbf{v}_2 \sin\theta\big), \qquad h\big(\mathbf{v}_1 \cos\theta + \mathbf{v}_2 \sin\theta\big) \times (a_3 - c) = \mathbf{0}. \tag{2}$$

Since $(a_3 - c) \perp (a_1 - a_2)$, the cross-product condition yields a unique θ:

$$\tan\theta = \frac{(a_3 - c)\cdot \mathbf{v}_2}{(a_3 - c)\cdot \mathbf{v}_1}. \tag{3}$$

The two maps are mutual inverses, so (a_1, a_2, a_3) and (f_l, f_r, θ) are in analytic bijection. □

4.2 Any Relationship Between In-Hand and Offhand Object Can Be Expressed by Keypoint Constraints

Proposition 1. *For any prescribed relative coordinate transform $T_{\text{true}} \in SE(3)$ between two rigid objects, there exist two non-parallel spatial line-coincidence constraints such that any unknown $T \in SE(3)$ satisfying the resulting system is uniquely determined, and the solution is T_{true}.*

Proof.
Existence. Place the objects at T_{true}. In the global frame pick two non-parallel lines ℓ_1, ℓ_2, with body-A parameters L_{A1}, L_{A2} and body-B parameters L_{B1}, L_{B2}. Rigid kinematics gives

$$T_{\text{true}}(L_{B1}) = L_{A1}, \qquad T_{\text{true}}(L_{B2}) = L_{A2}. \tag{4}$$

Hence such a pair of characteristic line correspondences exists.

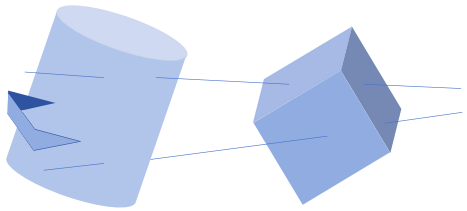

Figure 5: Two non-parallel line coincidences fix a unique $T \in SE(3)$.

Reduction. Let $T \in SE(3)$ satisfy the same two coincidences. Align local frames so that the Z axis of A coincides with L_{A1} and the Z axis of B with L_{B1}. Then $T(L_{B1}) = L_{A1}$ forces T to preserve the Z axis, so T retains two degrees of freedom: a rotation θ about Z and a translation d along Z. For $\mathbf{p} = (x, y, z)^\top$ on B, the image $\mathbf{p}' = (x', y', z')^\top$ is $x' = x\cos\theta - y\sin\theta,\ y' = x\sin\theta + y\cos\theta,\ z' = z + d$,

$$x' = x\cos\theta - y\sin\theta, \quad y' = x\sin\theta + y\cos\theta, \quad z' = z + d. \tag{5}$$

Unique θ. The second constraint is $T(L_{B2}) = L_{A2}$. Non-parallelism implies $L_{B2} \nparallel Z$. Let $\mathbf{v}_{xy} = (v_x, v_y)^\top \neq \mathbf{0}$ be the XY projection of the direction of L_{B2}, and $\mathbf{u}_{xy} = (u_x, u_y)^\top$ that of L_{A2}. Rotation about Z preserves the projected length, $\|\mathbf{v}_{xy}\| = \|\mathbf{u}_{xy}\|$, and

$$\begin{pmatrix} \cos\theta & -\sin\theta \\ \sin\theta & \cos\theta \end{pmatrix} \begin{pmatrix} v_x \\ v_y \end{pmatrix} = \begin{pmatrix} u_x \\ u_y \end{pmatrix}. \tag{6}$$

The determinant is nonzero, so this system has a unique real solution $\theta \in [0, 2\pi)$.

Unique d. Pick a known point $\mathbf{p}_0 = (x_0, y_0, z_0)^\top$ on L_{B2}. Substituting x_0, y_0 and the solved θ into (5) yields constants x_0', y_0'. The image of $\mathbf{p}_0$ lies on L_{A2}, and $L_{A2} \nparallel Z$, so (x_0', y_0') determines a unique height Z_{target} on L_{A2}. The third identity in (5) gives $z_0 + d = Z_{\text{target}}$, hence $d = Z_{\text{target}} - z_0$.

Thus T is uniquely solved and equals T_{true}. □

5 Experiments

5.1 RoboHarness can outperform VLA and WAM across tasks

BEHAVIOR 2025: Long-horizon household tasks. We compare against state-of-the-art VLAs such as Galaxea G0.5 and $\pi_{0.5}$, evaluating both success rate and subgoal completion rate (Liu et al., 2026; Black et al., 2025a). We additionally include the first- and

second-place winners of the BEHAVIOR Challenge 2025 (RLC and Comet, respectively) (Larchenko et al., 2025; Bai et al., 2025), whose q-scores are reported on the official leaderboard. The RoboHarness column is Qwen3.8-Flash-Next. Each entry is the mean q-score over five instances per task. Due to the limited evaluation budget, we only run five instances per task with seed `20260911`, using slots $0, 3, 5, 7, 9$ (instance IDs $301, 304, 306, 308, 310$). The maximum tick count is set separately for each task, and the ten limits are listed in Appendix A. Per-instance scores and the rollout archive are in Appendix A.

Table 3: BEHAVIOR Challenge 2025 task results. The RoboHarness column is Qwen3.8-Flash-Next, and each entry is the five-instance mean in Appendix A.

ID	Task	RLC	Comet	π_0.5 4ep	G0.5 1ep	RoboHarness
0	turning on radio	0.6000	1.0000	0.1500	0.3000	0.4000
1	picking up trash	0.6667	0.7667	0.5500	0.8167	0.8667
2	putting away Halloween decorations	0.2000	0.5000	0.3714	0.4857	0.6857
3	cleaning up plates and food	0.1857	0.0000	0.3857	0.1786	0.2571
4	can meat	0.0000	0.0000	0.0333	0.0111	0.0000
5	setting mousetraps	0.3333	0.0000	0.1083	0.5083	0.8333
6	hiding Easter eggs	0.1444	0.2444	0.0778	0.0500	0.4222
7	picking up toys	0.3000	0.0000	0.1833	0.3167	0.3667
8	rearranging kitchen furniture	0.3000	0.3750	0.3875	0.3500	0.4000
9	putting up Christmas decorations inside	0.4333	0.0000	0.5611	0.4667	0.2222
Percentage		31.635%	28.861%	28.084%	34.838%	44.539%

5.2 Does the RoboHarness Generalize across Objects?

Objects. The catalog is the 100 rigid objects in Figure 6. Assets are taken from the BEHAVIOR object library (Li et al., 2024), have extent at most 0.21 m, and have mass at most 1.2 kg. `saucepan` and one recorded outlier are kept outside those two cuts.

Tasks. The task is to pick up each of the 100 objects, and to place each object in the embodiment and simulation environment on which the corresponding model was trained. For ASPIRE and RoboHarness, the object is placed on the floor and the robot starts from a stooping pose. For models trained on LIBERO (Liu et al., 2023), the LIBERO tabletop scene is cleared and the object is placed at the center of the table. Success conditions:

Max ticks	2000
Success	$\max z - z_0 \geq 5\,\text{cm}$



Figure 6: The 100 objects used to evaluate generalization across objects. The catalog is drawn from the BEHAVIOR object library.

Analysis and results.

Generalization is not positively correlated with the amount of robot data. In Figure 7, Faster-WAM reaches 61/100 from a few hours of robot demonstrations, while several models trained on thousands of hours score lower (Zhao et al., 2026). The three world action models initialized from Wan2.2, Faster-WAM, OpenWAM-α, and LingBot-VA, sit above most vision-language-action models despite very different robot-data budgets (Wan Team, 2025; Wang et al., 2026; Li et al., 2026).

Multimodal capabilities matter more than model parameter count. The RoboHarness points use no robot training set and still score 68–86. Qwen3.8-27B is counted only on the objects that have finished, which gives 81 successes, and unfinished objects are left out of that score. These points sit above the learned policies. Among those policies, the smaller world action models outscore larger vision-language-action models such as OpenVLA-OFT (Kim et al., 2025a). Per-object outcomes are in Appendix D.

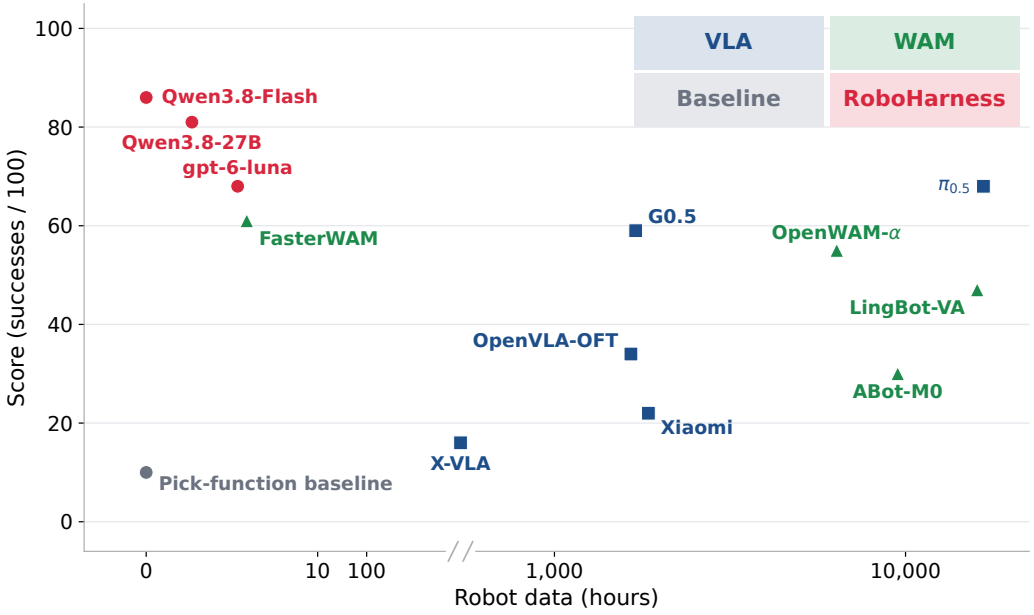

Figure 7: Successes out of 100 objects versus estimated hours of robot data. Colors: VLA (blue), world action model (green), RoboHarness (red), and the ASPIRE-pick baseline (gray). The hour conversion is in Appendix C.

6 Discussion

This work explores a new technical path for generalist Embodied AI, constructing a simple but effective harness design that lets an LLM directly command a robot, and experimentally shows that this approach substantially outperforms both VLA and WAM baselines in both performance and ceiling. An LLM already has good ideas for embodied tasks—for instance, picking up a book to pull a harder-to-reach object closer—but does not know how to physically realize those motions. RoboHarness bridges this gap with handcraft skills and prompts. Future work aims to close this gap further by training the model to become more physically grounded.

AI use statement

Generative AI tools were used in three ways. (i) They assisted writing: drafting the manuscript text from author-provided outlines and Chinese prose. (ii) They assisted running the experiments: launching jobs, monitoring long rollouts, and flagging failed runs so that each experiment kept running stably, while every experimental setting and every reported number was checked by the authors. (iii) They assisted coding: writing boilerplate and debugging RoboHarness system. All AI-assisted text was refined by the authors against the source outline, and every AI-assisted experimental result was checked by the authors against the underlying evaluator outputs.

References

Michael Ahn, Anthony Brohan, Noah Brown, Yevgen Chebotar, Omar Cortes, Byron David, Chelsea Finn, Chuyuan Fu, Keerthana Gopalakrishnan, Karol Hausman, et al. Do as I can, not as I say: Grounding language in robotic affordances. In *Proceedings of the 6th Conference on Robot Learning*, volume 205 of *Proceedings of Machine Learning Research*, pp. 287–318. PMLR, 2023. URL https://proceedings.mlr.press/v205/ichter23a.html.

Junjie Bai, Yu-Wei Chao, Qizhi Chen, Jinwei Gu, Moo Jin Kim, Zhaoshuo Li, Xuan Li, Tsung-Yi Lin, Ming-Yu Liu, Nic Ma, Kaichun Mo, Delin Qu, Shangkun Sun, Hongchi Xia, Fangyin Wei, and Xiaohui Zeng. OpenPI Comet: Competition solution for 2025 BEHAVIOR challenge. *arXiv preprint arXiv:2512.10071*, 2025. URL https://arxiv.org/abs/2512.10071.

Johan Bjorck, Fernando Castañeda, Nikita Cherniadev, Xingye Da, Runyu Ding, Linxi Jim Fan, Yu Fang, Dieter Fox, et al. GR00T N1: An open foundation model for generalist humanoid robots. *arXiv preprint arXiv:2503.14734*, 2025. URL https://arxiv.org/abs/2503.14734.

Kevin Black, Noah Brown, James Darpinian, Karan Dhabalia, Danny Driess, Adnan Esmail, Michael Equi, Chelsea Finn, Niccolo Fusai, Manuel Y. Galliker, et al. $\pi_{0.5}$: a vision-language-action model with open-world generalization. In *Proceedings of the 9th Conference on Robot Learning*, volume 305 of *Proceedings of Machine Learning Research*, pp. 17–40. PMLR, 2025a. URL https://proceedings.mlr.press/v305/black25a.html.

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 *Proceedings of Robotics: Science and Systems*, Los Angeles, CA, USA, 2025b. URL https://arxiv.org/abs/2410.24164.

Anthony Brohan, Noah Brown, Justice Carbajal, Yevgen Chebotar, Joseph Dabis, Chelsea Finn, Chuyuan Fu, Keerthana Gopalakrishnan, et al. RT-2: Vision-language-action models transfer web knowledge to robotic control. In *Proceedings of the 7th Conference on Robot Learning*, volume 229 of *Proceedings of Machine Learning Research*, pp. 2165–2183. PMLR, 2023. URL https://proceedings.mlr.press/v229/zitkovich23a.html.

Qingwen Bu, Jisong Cai, Li Chen, et al. AgiBot World Colosseo: A large-scale manipulation platform for scalable and intelligent embodied systems. In *2025 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2025. doi: 10.1109/IROS60139.2025.11247088.

Rui Cai, Jun Guo, Xinze He, Piaopiao Jin, Jie Li, Bingxuan Lin, Futeng Liu, Wei Liu, Fei Ma, Kun Ma, Feng Qiu, Heng Qu, Yifei Su, Qiao Sun, Dong Wang, Donghao Wang, Yunhong Wang, Rujie Wu, Diyun Xiang, Yu Yang, Hangjun Ye, Yuan Zhang, and Quanyun Zhou. Xiaomi-Robotics-0: An open-sourced vision-language-action model with real-time execution. *arXiv preprint arXiv:2602.12684*, 2026. URL https://arxiv.org/abs/2602.12684.

Tianxing Chen, Yue Chen, Zixuan Li, Junyuan Tang, Kailun Su, et al. RoboDojo: A unified sim-and-real benchmark for comprehensive evaluation of generalist robot manipulation policies. *arXiv preprint arXiv:2607.04434*, 2026. URL https://arxiv.org/abs/2607.04434.

Danny Driess, Jost Tobias Springenberg, Brian Ichter, Lili Yu, Adrian Li-Bell, Karl Pertsch, Allen Z. Ren, Homer Walke, Quan Vuong, Lucy Xiaoyang Shi, and Sergey Levine. Knowledge insulating vision-language-action models: Train fast, run fast, generalize better. In *Advances in Neural Information Processing Systems*, 2025. URL https://arxiv.org/abs/2505.23705.

Kuan Fang, Fangchen Liu, Pieter Abbeel, and Sergey Levine. MOKA: Open-world robotic manipulation through mark-based visual prompting. In *Proceedings of Robotics: Science and Systems*, 2024. URL https://arxiv.org/abs/2403.03174.

Letian Fu, Justin Yu, Karim El-Refai, Ethan Kou, Haoru Xue, Huang Huang, Wenli Xiao, Guanzhi Wang, Dantong Niu, Li Fei-Fei, Guanya Shi, Jiajun Wu, Shankar Sastry, Yuke Zhu, Ken Goldberg, and Linxi Jim Fan. CaP-X: A framework for benchmarking and improving coding agents for robot manipulation. *arXiv preprint arXiv:2603.22435*, 2026. URL https://arxiv.org/abs/2603.22435.

Haoxu Huang, Fanqi Lin, Yingdong Hu, Shengjie Wang, and Yang Gao. CoPa: General robotic manipulation through spatial constraints of parts with foundation models. In *2024 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pp. 9488–9495, 2024. doi: 10.1109/IROS58592.2024.10801352.

Wenlong Huang, Chen Wang, Ruohan Zhang, Yunzhu Li, Jiajun Wu, and Li Fei-Fei. Voxposer: Composable 3D value maps for robotic manipulation with language models. In *Proceedings of the 7th Conference on Robot Learning*, volume 229 of *Proceedings of Machine Learning Research*, pp. 540–562. PMLR, 2023a. URL https://proceedings.mlr.press/v229/huang23b.html.

Wenlong Huang, Fei Xia, Ted Xiao, Harris Chan, Jacky Liang, Pete Florence, Andy Zeng, Jonathan Tompson, Igor Mordatch, Yevgen Chebotar, Pierre Sermanet, Tomas Jackson, Noah Brown, Linda Luu, Sergey Levine, Karol Hausman, and Brian Ichter. Inner monologue: Embodied reasoning through planning with language models. In *Proceedings of the 6th Conference on Robot Learning*, volume 205 of *Proceedings of Machine Learning Research*, pp. 1769–1782. PMLR, 2023b. URL https://proceedings.mlr.press/v205/huang23c.html.

Wenlong Huang, Chen Wang, Yunzhu Li, Ruohan Zhang, and Li Fei-Fei. ReKep: Spatio-temporal reasoning of relational keypoint constraints for robotic manipulation. In *Proceedings of the 8th Conference on Robot Learning*, volume 270 of *Proceedings of Machine Learning Research*, pp. 4573–4602. PMLR, 2025. URL https://proceedings.mlr.press/v270/huang25g.html.

Tao Jiang, Tianyuan Yuan, Yicheng Liu, Chenhao Lu, Jianning Cui, Xiao Liu, Shuiqi Cheng, Jiyang Gao, Huazhe Xu, and Hang Zhao. Galaxea Open-World dataset and G0 dual-system VLA model. *arXiv preprint arXiv:2509.00576*, 2025. URL https://arxiv.org/abs/2509.00576.

Alexander Khazatsky, Karl Pertsch, Suraj Nair, Ashvin Balakrishna, Sudeep Dasari, Siddharth Karamcheti, Soroush Nasiriany, Mohan Kumar Srirama, Lawrence Yunliang Chen, Kirsty Ellis, et al. DROID: A large-scale in-the-wild robot manipulation dataset. In *Proceedings of Robotics: Science and Systems*, 2024. URL https://arxiv.org/abs/2403.12945.

Moo Jin Kim, Chelsea Finn, and Percy Liang. Fine-tuning vision-language-action models: Optimizing speed and success. In *Proceedings of Robotics: Science and Systems*, Los Angeles, CA, USA, 2025a. doi: 10.15607/RSS.2025.XXI.017.

Moo Jin Kim, Karl Pertsch, Siddharth Karamcheti, Ted Xiao, Ashwin Balakrishna, Suraj Nair, Rafael Rafailov, Ethan Foster, Grace Lam, Pannag Sanketi, et al. OpenVLA: An open-source vision-language-action model. In *Proceedings of the 8th Conference on Robot Learning*, volume 270 of *Proceedings of Machine Learning Research*. PMLR, 2025b. URL https://arxiv.org/abs/2406.09246.

Ilia Larchenko, Gleb Zarin, and Akash Karnatak. Task adaptation of vision-language-action model: 1st place solution for the 2025 BEHAVIOR challenge. *arXiv preprint arXiv:2512.06951*, 2025. URL https://arxiv.org/abs/2512.06951.

Chengshu Li, Ruohan Zhang, Josiah Wong, Cem Gokmen, Sanjana Srivastava, Roberto Martín-Martín, Chen Wang, Gabrael Levine, Wensi Ai, Benjamin Martinez, et al. BEHAVIOR-1K: A human-centered, embodied AI benchmark with 1,000 everyday activities and realistic simulation. *arXiv preprint arXiv:2403.09227*, 2024. URL https://arxiv.org/abs/2403.09227.

Lin Li, Qihang Zhang, Yiming Luo, Shuai Yang, Ruilin Wang, Fei Han, Mingrui Yu, Zelin Gao, Nan Xue, Xing Zhu, Yujun Shen, and Yinghao Xu. LingBot-VA: Causal world modeling for robot control. *arXiv preprint arXiv:2601.21998*, 2026. URL https://arxiv.org/abs/2601.21998.

Yi Li, Yuquan Deng, Jesse Zhang, Joel Jang, Marius Memmel, Raymond Yu, Caelan Reed Garrett, Fabio Ramos, Dieter Fox, Anqi Li, Abhishek Gupta, and Ankit Goyal. HAMSTER: Hierarchical action models for open-world robot manipulation. In *International Conference on Learning Representations*, 2025. URL https://arxiv.org/abs/2502.05485.

Jacky Liang, Wenlong Huang, Fei Xia, Peng Xu, Karol Hausman, Brian Ichter, Pete Florence, and Andy Zeng. Code as policies: Language model programs for embodied control. In *2023 IEEE International Conference on Robotics and Automation (ICRA)*, pp. 9493–9500, 2023. doi: 10.1109/ICRA48891.2023.10160591.

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*, 2023. URL https://arxiv.org/abs/2306.03310.

Yicheng Liu, Zibin Dong, Baijun Ye, Tianyuan Yuan, Tao Jiang, Anqi Yang, Shicheng Cao, Haonan Liu, Yue Sun, Zihan Guo, et al. G0.5: One autoregressive stream for robot reasoning and action. *arXiv preprint arXiv:2608.11739*, 2026. URL https://arxiv.org/abs/2608.11739.

Runyu Lu, Yubo Wu, Ethan Kou, Letian Fu, Wenli Xiao, Ajay Mandlekar, Yinzhen Xu, Guanya Shi, Ken Goldberg, Ang Chen, Mosharaf Chowdhury, Yuke Zhu, Linxi Jim Fan, and Guanzhi Wang. ASPIRE: Agentic /skills discovery for robotics. *arXiv preprint arXiv:2607.00272*, 2026. URL https://arxiv.org/abs/2607.00272.

Soroush Nasiriany, Fei Xia, Wenhao Yu, Ted Xiao, Jacky Liang, Ishita Dasgupta, Annie Xie, Danny Driess, Ayzaan Wahid, Zhuo Xu, Quan Vuong, Tingnan Zhang, Tsang-Wei Edward Lee, Kuang-Huei Lee, Peng Xu, Sean Kirmani, Yuke Zhu, Andy Zeng, Karol Hausman, Nicolas Heess, Chelsea Finn, Sergey Levine, and Brian Ichter. PIVOT: Iterative visual prompting elicits actionable knowledge for VLMs. In *Proceedings of the 41st International Conference on Machine Learning*, volume 235 of *Proceedings of Machine Learning Research*, pp. 37321–37341. PMLR, 2024. URL https://proceedings.mlr.press/v235/nasiriany24a.html.

Noematrix Team. RoboRSI. https://robo-rsi.com/blog/2-roborsi/, 2026. Research preview.

Open X-Embodiment Collaboration. Open X-Embodiment: Robotic learning datasets and RT-X models. In *2024 IEEE International Conference on Robotics and Automation (ICRA)*, pp. 6892–6903, 2024. doi: 10.1109/ICRA57147.2024.10611477.

OpenAI RoboCurve. GPT-6 Astra. https://openai.robocurve.org/gpt-6-astra/, 2026. RoboCurve evaluation page.

Mingjie Pan, Jiyao Zhang, Tianshu Wu, Yinghao Zhao, Wenlong Gao, and Hao Dong. OmniManip: Towards general robotic manipulation via object-centric interaction primitives as spatial constraints. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 17359–17369, 2025. doi: 10.1109/CVPR52734.2025.01618.

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. In *Proceedings of Robotics: Science and Systems*, Los Angeles, CA, USA, 2025. doi: 10.15607/RSS.2025.XXI.012.

Physical Intelligence. $\pi_{0.7}$: a steerable generalist robotic foundation model with emergent capabilities. *arXiv preprint arXiv:2604.15483*, 2026. URL https://arxiv.org/abs/2604.15483.

Qwen Team. Qwen3.5: Towards native multimodal agents. https://qwen.ai/blog?id=qwen3.5, February 2026a. Official model release; no standalone arXiv report as of this submission.

Qwen Team. Qwen3.8-Flash-Next: A new architecture, towards ultimate cost-efficiency. https://qwen.ai/blog?id=qwen3.8-flash-next, August 2026b.

Junyao Shi, Rujia Yang, Kaitian Chao, Selina Bingqing Wan, Yifei Shao, Jiahui Lei, Jianing Qian, Long Le, Pratik Chaudhari, Kostas Daniilidis, Chuan Wen, and Dinesh Jayaraman. Maestro: Orchestrating robotics modules with vision-language models for zero-shot generalist robots. *arXiv preprint arXiv:2511.00917*, 2025. URL https://arxiv.org/abs/2511.00917. Withdrawn.

Yang Tian, Yuyin Yang, Yiman Xie, Zetao Cai, Xu Shi, Ning Gao, Hangxu Liu, Xuekun Jiang, Zherui Qiu, Feng Yuan, Yaping Li, Ping Wang, Junhao Cai, Jia Zeng, Hao Dong, and Jiangmiao Pang. InternData-A1: Pioneering high-fidelity synthetic data for pre-training generalist policy. *arXiv preprint arXiv:2511.16651*, 2025. URL https://arxiv.org/abs/2511.16651.

Wan Team. Wan2.2. https://github.com/Wan-Video/Wan2.2, 2025. Model card and code release. The technical report is Wan et al. (2025).

Wan Team, Ang Wang, Baole Ai, Bin Wen, Chaojie Mao, Chen-Wei Xie, Di Chen, Feiwu Yu, Haiming Zhao, Jianxiao Yang, Jianyuan Zeng, et al. Wan: Open and advanced large-scale video generative models. *arXiv preprint arXiv:2503.20314*, 2025. URL https://arxiv.org/abs/2503.20314.

Yuran Wang, Siqiao Huang, Mingleyang Li, Chenhao Zhang, Jiaqi Liang, Weiyang Jin, Yue Chen, Xuemin Chi, Donghao Zhou, Qize Yu, et al. OpenWAM: An open, modular exploration towards systematic world-action model pretraining. *arXiv preprint arXiv:2609.07398*, 2026. URL https://arxiv.org/abs/2609.07398.

Kun Wu, Chengkai Hou, Jiaming Liu, Zhengping Che, Xiaozhu Ju, et al. RoboMIND: Benchmark on multi-embodiment intelligence normative data for robot manipulation. In *Proceedings of Robotics: Science and Systems*, Los Angeles, CA, USA, 2025a. doi: 10.15607/RSS.2025.XXI.152.

Shihan Wu, Xuecheng Liu, Shaoxuan Xie, Pengwei Wang, Xinghang Li, et al. RoboCOIN: An open-sourced bimanual robotic data collection for integrated manipulation. *arXiv preprint arXiv:2511.17441*, 2025b. URL https://arxiv.org/abs/2511.17441.

Yandan Yang, Shuang Zeng, Tong Lin, Xinyuan Chang, Dekang Qi, Junjin Xiao, Haoyun Liu, Ronghan Chen, Yuzhi Chen, Dongjie Huo, Feng Xiong, Xing Wei, Zhiheng Ma, and Mu Xu. ABot-M0: VLA foundation model for robotic manipulation with action manifold learning. *arXiv preprint arXiv:2602.11236*, 2026. URL https://arxiv.org/abs/2602.11236.

Tianyuan Yuan. LIBERO MuJoCo 3.3.2 for FastWAM. https://huggingface.co/datasets/yuanty/LIBERO-fastwam, 2026. LeRobot release used by the Faster-WAM training scripts.

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 *Proceedings of the 8th Conference on Robot Learning*, volume 270 of *Proceedings of Machine Learning Research*, pp. 4005–4020. PMLR, 2025. URL https://proceedings.mlr.press/v270/yuan25c.html.

Yixian Zhang, Huanming Zhang, Feng Gao, Xiao Li, Zhihao Liu, Yi Nie, Chunyang Zhu, Jiaxing Qiu, Yuchen Yan, Jiyuan Liu, Wenhao Tang, Jiaji Rao, Zhengru Fang, Changxu Wei, Yu Wang, Wenbo Ding, and Chao Yu. Harness VLA: Steering frozen VLAs into reliable manipulation primitives via memory-guided agents. *arXiv preprint arXiv:2607.08448*, 2026. URL https://arxiv.org/abs/2607.08448.

Weiheng Zhao, Haoyi Jiang, Xin Shi, Liu Liu, Fan Huang, Zhizhong Su, Wei Sui, and Xinggang Wang. Faster-WAM: Efficient inference-time future conditioning for robust world action models. *arXiv preprint arXiv:2608.04404*, 2026. URL https://arxiv.org/abs/2608.04404.

Jinliang Zheng, Jianxiong Li, Zhihao Wang, Dongxiu Liu, Xirui Kang, Yuchun Feng, Yinan Zheng, Jiayin Zou, Yilun Chen, Jia Zeng, et al. X-VLA: Soft-prompted transformer as scalable cross-embodiment vision-language-action model. In *International Conference on Learning Representations*, 2026. URL https://arxiv.org/abs/2510.10274.

A BEHAVIOR Five-Instance Scores

The RoboHarness column of Table 3 is a five-instance sample of each of the first ten BEHAVIOR Challenge 2025 tasks, and every rollout in that column uses Qwen3.8-Flash-Next. The robot is the R1 Pro embodiment of the challenge. The sample is fixed by seed `20260911`. Every task loads the same evaluation slots $0, 3, 5, 7, 9$, which are activity instances $301, 304, 306, 308, 310$. Slots $1, 2, 4, 6, 8$ (instances $302, 303, 305, 307, 309$) are outside this sample. Each instance is one official rollout. The prompt is specific to each task but shared across instances, and is written by a human. It includes a procedure for accomplishing the task. The official BEHAVIOR Challenge 2025 evaluator scores the rollout, and a cell in Table 5 is `q_score.final` from that evaluation. The task mean is the unweighted average of those five scores, rounded to four decimals, and is the number reported in Table 3.

An episode ends when the task goal is already satisfied or when the simulator reaches the maximum tick count of that task. The maximum is the official 2025 limit for the task, set to twice the mean length of its human demonstrations, and it is counted in simulator control steps. Table 4 lists the limit used for each task.

Table 4: Official maximum tick count for each BEHAVIOR Challenge 2025 task in this sample. The episode stops at this count unless the goal is already satisfied.

ID	Task	Max ticks
0	turning on radio	4299
1	picking up trash	10535
2	putting away Halloween decorations	27664
3	cleaning up plates and food	27392
4	can meat	23694
5	setting mousetraps	20343
6	hiding Easter eggs	15239
7	picking up toys	37781
8	rearranging kitchen furniture	17886
9	putting up Christmas decorations inside	27437

Table 5: Per-instance q-score for the five-instance RoboHarness sample. Slots $0, 3, 5, 7, 9$ are instances $301, 304, 306, 308, 310$.

ID	Task	301 slot 0	304 slot 3	306 slot 5	308 slot 7	310 slot 9	Mean
0	turning on radio	0.0000	0.0000	0.0000	1.0000	1.0000	0.4000
1	picking up trash	1.0000	0.3333	1.0000	1.0000	1.0000	0.8667
2	putting away Halloween decorations	0.5714	0.7143	0.4286	0.7143	1.0000	0.6857
3	cleaning up plates and food	0.2857	0.2857	0.2857	0.1429	0.2857	0.2571
4	can meat	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000
5	setting mousetraps	1.0000	1.0000	0.5000	1.0000	0.6667	0.8333
6	hiding Easter eggs	1.0000	0.0000	0.7778	0.3333	0.0000	0.4222
7	picking up toys	0.0000	0.6667	0.5000	0.6667	0.0000	0.3667
8	rearranging kitchen furniture	0.2500	0.5000	0.2500	0.5000	0.5000	0.4000
9	putting up Christmas decorations inside	0.3333	0.2222	0.3333	0.1111	0.1111	0.2222
Task 0–9 mean							0.44539

B Full Embodied Skill Catalog

C Evaluation Setting and Robot-Data Hours

Test. Each learned policy is evaluated on a cleared LIBERO table. The object is placed at the center, the prompt is `pick up the {name}`, and the trial runs for 2000 steps. The

Table 6: Embodied skills shipped with the `embodied_claude_code` plugin, excluding the trivial single-object skills `close-box`, `cut-object`, and `stand-trash-can-upright`. Each row is condensed to one sentence describing what the skill does.

Skill	Description
`behavior-v2-baseline`	contract of scene perception, task decomposition, primitive selection, and execution monitoring.
`navigate-to-target`	Drives the chassis toward a marked target.
`open-doors-and-drawers`	Open hinged doors or drawers.
`pick-up-object`	Plans and executes a grasp of a visible object.
`place-object-in-container`	Aligns a held object over an open container and releases it only after the object has settled inside.
`traverse-narrow-passages`	Drives chassis through tight doorway .

trial succeeds when the target root rises by at least 5 cm. Qwen3.8-Flash and gpt-6-luna use the RoboHarness on the R1 Pro, with the object on the floor and the same success rule. An hour counts robot recordings and human egocentric manipulation video once. Web-video generation and VQA are not counted.

Qwen3.8-Flash.

- The run calls the served model `Qwen3.8-Flash-Next-FP8` with the skill prompt. This model has no released robot checkpoint on Hugging Face.
- The run has no robot manipulation training set, so the robot-data hours are 0.

GPT-6-luna.

- The run calls the served model `gpt-6-luna` with the skill prompt. This model has no released robot checkpoint on Hugging Face.
- The run has no robot manipulation training set, so the robot-data hours are 0.

$\pi_{0.5}$.

- The run uses the released `pi05_libero` checkpoint from the openpi release. The policy replans every 5 actions.
- The paper states about 400 hours of mobile-manipulator data, and states that 97.6% of the examples in the first training phase come from other robots or the web (Black et al., 2025a). Converting the whole first-phase mixture by that share gives 16,666.7 hours. The paper does not separate the web portion from the other robot data.

Faster-WAM.

- The run uses `hustvl/FasterWAM`, the released LIBERO weights at step 21700. The policy replans every 10 actions.
- The paper initializes the model from Wan2.2-5B, marks embodied pretraining as absent, and trains on four LIBERO suites of 500 demonstrations for 10 epochs (Zhao et al., 2026; Wan Team, 2025). The training release `yuanty/LIBERO-fastwam` states 1,712 episodes and 277,713 frames at 20 frames per second, which converts to 3.86 hours (Yuan, 2026). The Wan corpus is web video and is not counted (Wan Team et al., 2025).

Galaxea G0.5.

- The run uses the `g05-libero` checkpoint released in `OpenGalaxea/G05`. The policy predicts 10 action steps at a time.
- The G0.5 paper states 14 embodiments and does not state a training duration (Liu et al., 2026). The estimate therefore uses stage 1 of the previous G0 paper, which states about

1,000 hours of Open X-Embodiment, 500 hours of Galaxea Open-World, and 200 hours collected in-house (Jiang et al., 2025; Open X-Embodiment Collaboration, 2024). The no-op-filtered LIBERO demonstrations used for this checkpoint convert to 3.80 hours at 20 Hz. Adding these durations gives 1,703.8 hours.

OpenWAM-α.

- The run uses `OpenWAM/OpenWAM-Alpha-Sim-LIBERO` at step 10690. Denoising uses 10 steps, and the inference horizon is 10 actions.
- The paper lists 1,442 hours of egocentric human video, 1,794 hours of AgiBot, 686 hours of RoboCOIN, 1,007 hours of DROID, and 1,440 hours of InternData-A1, and states that this mixture is seen for one epoch (Wang et al., 2026; Wu et al., 2025b; Tian et al., 2025). Adding these durations gives 6,369 hours. The Wan2.2 initialization is web video and is not counted (Wan Team et al., 2025; Wan Team, 2025).

LingBot-VA.

- The run uses `robbyant/lingbot-va-posttrain-libero-long`. The trial uses the shared setting of 2000 steps and a 5 cm lift.
- The paper states approximately 16K hours of robot manipulation data, and states that pretraining consumes 1.4T tokens (Li et al., 2026). The plotted value is the stated 16,000 hours. The token count records how much data the run consumes, and it is not added again as a second copy of the corpus.

OpenVLA-OFT.

- The run uses `moojink/openvla-7b-oft-finetuned-libero-10`. The unnormalization key is `libero_10_no_noops`, and the client executes the full 8-step action chunk.
- The OpenVLA paper states 970k demonstrations from 27 Open X-Embodiment datasets, trained for 27 epochs, with the DROID portion removed for the last third of training (Kim et al., 2025b; Open X-Embodiment Collaboration, 2024). The released frame counts of those 27 datasets convert to 1,651.4 hours. The OFT paper fine-tunes this checkpoint on LIBERO and does not add another pretraining corpus (Kim et al., 2025a).

ABot-M0.

- The run uses `acvlab/ABot-M0-LIBERO` at step 40000. The action chunk is 10 steps, and the unnormalization key is `franka`.
- The paper states that UniACT contains more than six million trajectories and 9,500 hours (Yang et al., 2026). The plotted value is the stated 9,500 hours.

Xiaomi-Robotics-0.

- The run uses `XiaomiRobotics/Xiaomi-Robotics-0-LIBERO`. The task id is `libero_all`, and the policy replans every 10 actions.
- The paper states about 200×10^6 robot timesteps on a unified 30 Hz timeline (Cai et al., 2026). These timesteps convert to 1,851.9 hours. The 338 hours and 400 hours of in-house tasks stated in the paper are already inside this total.

X-VLA.

- The run uses `2toINF/X-VLA-Libero`. The domain id is 3, and denoising uses 10 steps.
- The pretraining table lists episode counts: AgiBot has 141K episodes, DROID is listed as two rows of 45K episodes, and RoboMIND is listed as 19K, 2K, 25K, and 11K episodes (Zheng et al., 2026). AgiBot states 1,001,552 trajectories and 2,976.4 hours, which converts to 419.0 hours for the listed 141K episodes (Bu et al., 2025). DROID states 76k successful trajectories and 350 hours, and the two 45K rows are one subset of that release, which converts to 207.2 hours (Khazatsky et al., 2024). RoboMIND states 107k

trajectories and 305.5 hours for the full corpus, which converts to 162.7 hours for the listed 57K episodes (Wu et al., 2025a). Adding these three durations gives 788.9 hours. The sampling weights in the X-VLA table are mixture weights, and they are not used as durations.

ASPIRE pick function.

- The baseline has no learned checkpoint. Each object is scored by one call of the ASPIRE pick function, and the same 5 cm lift is a success (Lu et al., 2026).
- The baseline has no robot manipulation training set, so the robot-data hours are 0.

D Per-Object Outcomes

Table 7 lists the twelve finished runs in Figure 7. A check is a success and a cross is a failure under the 5 cm rule of Section 5.2.

Table 7: Per-object outcome on the 100 objects. A check is a success under the 5 cm rule and a cross is a failure. Columns are Qwen3.8-Flash, gpt-6-luna, $\pi_{0.5}$, Faster-WAM, G0.5, OpenWAM-α, LingBot-VA, OpenVLA-OFT, ABot-M0, Xiaomi-Robotics-0, X-VLA, and the ASPIRE pick function.

#	Object	Flash	luna	$\pi_{0.5}$	F-WAM	G0.5	OWAM	Ling	OVLA	ABot	Xiao	XVLA	Pick
1	alarm_clock	×	×	×	×	×	×	×	×	×	×	×	×
2	apple	✓	✓	✓	✓	✓	✓	✓	×	✓	×	×	×
3	avocado	✓	✓	✓	✓	✓	×	✓	×	×	✓	×	×
4	banana	✓	✓	×	✓	×	×	×	×	×	×	✓	×
5	bar_soap	×	×	✓	✓	✓	✓	×	×	×	×	×	×
6	baseball	✓	×	✓	✓	✓	✓	✓	✓	✓	✓	✓	×
7	battery	✓	✓	×	×	×	✓	✓	✓	×	×	×	×
8	bell_pepper	✓	✓	✓	✓	✓	✓	×	✓	✓	✓	×	×
9	blackboard_eraser	✓	✓	✓	✓	✓	✓	✓	×	✓	✓	×	×
10	calculator	×	×	×	×	✓	×	×	×	×	×	×	×
11	can_of_soda	✓	✓	✓	✓	✓	✓	×	×	×	×	×	✓
12	canned_food	✓	×	✓	✓	×	✓	✓	✓	×	✓	✓	×
13	carrot	✓	×	✓	✓	✓	✓	✓	✓	✓	✓	×	×
14	cell_phone	✓	×	✓	×	×	×	×	×	×	×	×	✓
15	cellulose_tape_dispenser	✓	✓	✓	✓	✓	✓	×	×	×	×	×	×
16	coffee_cup	×	×	✓	✓	✓	✓	✓	×	✓	✓	✓	×
17	cucumber	✓	✓	✓	✓	✓	✓	×	×	✓	✓	×	×
18	deodorant_stick	✓	✓	✓	✓	✓	×	✓	×	×	×	×	✓
19	dental_floss	✓	✓	×	✓	×	✓	×	×	×	×	×	×
20	digital_camera	✓	×	×	×	×	×	×	×	×	×	×	×
21	duct_tape	✓	✓	×	×	×	×	×	×	×	×	×	✓
22	easter_egg	✓	✓	✓	✓	×	✓	✓	✓	×	×	×	×
23	egg	×	✓	✓	✓	✓	✓	✓	✓	✓	×	×	×
24	eggplant	✓	✓	✓	✓	✓	✓	×	×	×	✓	×	×
25	garlic	✓	✓	✓	✓	✓	✓	✓	✓	×	×	✓	×
26	gaming_controller	✓	✓	×	×	×	×	×	×	×	×	×	×
27	hockey_puck	✓	×	✓	✓	×	✓	✓	×	×	×	×	×
28	inhaler	✓	×	×	×	✓	✓	✓	✓	×	×	×	×
29	lemon	✓	✓	✓	✓	✓	✓	×	✓	✓	×	×	×
30	lime	✓	✓	✓	✓	✓	✓	✓	✓	✓	×	×	×
31	lipstick	✓	✓	✓	✓	✓	×	×	✓	×	×	×	×
32	mouse	✓	✓	✓	✓	✓	×	×	×	✓	×	×	×
33	mug	✓	✓	✓	✓	✓	✓	✓	×	×	×	×	✓
34	orange	✓	✓	✓	✓	×	✓	✓	✓	✓	×	×	×
35	paper_cup	✓	×	✓	×	✓	✓	×	×	×	×	✓	×
36	peach	✓	✓	✓	×	×	×	✓	✓	✓	×	×	×
37	pear	✓	✓	✓	✓	✓	✓	×	×	✓	×	×	×
38	pepper_grinder	✓	✓	✓	×	✓	✓	✓	×	×	✓	×	×
39	pill_bottle	✓	×	✓	✓	×	✓	✓	✓	×	×	×	×
40	pillar_candle	✓	×	×	×	×	×	×	×	×	×	×	×
41	plum	✓	✓	✓	✓	×	✓	✓	✓	×	×	×	×
42	pool_ball	✓	✓	✓	✓	✓	✓	✓	✓	✓	×	×	×
43	portable_game_console	✓	×	×	×	✓	×	✓	×	×	×	×	×
44	potato	✓	✓	✓	✓	✓	✓	×	✓	✓	×	✓	×
45	salt_shaker	✓	✓	✓	✓	✓	✓	✓	✓	×	×	×	×
46	soda_cup	✓	✓	✓	×	×	×	×	×	✓	×	✓	✓
47	stapler	✓	✓	✓	✓	✓	×	×	×	×	✓	×	×
48	tennis_ball	×	✓	✓	✓	✓	✓	✓	✓	✓	×	×	×
49	toy_dice	✓	✓	×	✓	×	✓	✓	✓	×	×	×	×
50	wallet	✓	×	✓	×	✓	×	×	×	×	×	×	×
51	artichoke	✓	✓	✓	✓	×	✓	✓	✓	×	×	×	×
52	baby_bottle	✓	✓	✓	✓	✓	✓	✓	×	×	×	×	×
53	beaker	✓	✓	✓	×	×	×	✓	×	×	×	✓	×
54	beefsteak_tomato	✓	✓	×	✓	✓	✓	×	×	✓	×	✓	×

#	Object	Flash	luna	$\pi_{0.5}$	F-WAM	G0.5	OWAM	Ling	OVLA	ABot	Xiao	XVLA	Pick
55	beer_glass	✓	✓	✓	×	×	✓	×	×	×	✓	×	×
56	beet	✓	×	×	×	×	×	×	×	×	×	×	×
57	bottle_of_hot_sauce	✓	×	✓	×	✓	×	✓	×	×	✓	×	×
58	bowl	✓	×	✓	×	×	×	×	×	×	×	×	✓
59	brussels_sprouts	✓	×	✓	✓	✓	✓	×	✓	×	×	×	×
60	can_of_beans	✓	×	✓	×	×	×	×	×	×	×	✓	×
61	can_of_sardines	✓	×	✓	×	✓	✓	×	×	×	×	✓	×
62	canteen	✓	✓	×	×	×	×	×	×	×	×	×	×
63	carton_of_milk	✓	✓	✓	✓	×	×	✓	×	×	×	×	×
64	fried_chicken	✓	✓	×	×	×	×	×	×	×	×	×	×
65	cinnamon_roll	✓	✓	✓	✓	✓	✓	✓	✓	×	×	×	×
66	marker	✓	×	✓	✓	✓	✓	×	✓	×	×	×	×
67	conch	✓	✓	×	×	×	×	×	×	×	×	×	×
68	corkscrew	✓	✓	×	✓	✓	×	✓	×	×	✓	×	×
69	gravy_boat	✓	✓	×	×	×	×	×	×	×	×	×	×
70	eyeglasses	×	×	×	×	×	×	×	×	×	×	×	✓
71	french_press	×	✓	×	×	×	×	×	×	×	×	×	×
72	gift_box	✓	✓	×	×	×	×	×	×	×	×	×	✓
73	glue_stick	✓	✓	×	✓	✓	✓	✓	✓	✓	✓	×	×
74	grapefruit	✓	✓	×	×	×	×	×	×	×	×	×	×
75	hard_drive	✓	✓	✓	✓	✓	×	×	×	×	×	×	×
76	hotdog	✓	✓	✓	×	✓	×	✓	×	×	×	×	×
77	spray_can	×	✓	✓	×	✓	×	×	×	×	×	×	×
78	jade_roller	×	✓	×	✓	✓	✓	✓	×	×	×	×	×
79	jar_of_jam	✓	✓	✓	✓	×	×	✓	×	✓	✓	✓	×
80	jigger	✓	✓	✓	✓	×	×	✓	×	×	×	×	×
81	light_bulb	✓	✓	×	✓	✓	✓	✓	×	×	×	×	×
82	lighter	✓	✓	×	×	✓	×	×	×	×	×	×	×
83	mango	✓	✓	✓	✓	✓	✓	×	✓	✓	×	×	×
84	match_box	✓	×	✓	✓	✓	✓	✓	✓	✓	✓	×	×
85	mousetrap	✓	✓	✓	✓	✓	×	✓	✓	×	×	×	×
86	pencil_case	✓	✓	✓	×	✓	×	×	×	×	×	×	×
87	plier	✓	×	×	✓	✓	✓	×	×	×	×	×	×
88	pomegranate	×	✓	✓	✓	✓	✓	✓	×	✓	×	✓	×
89	razor	×	×	×	✓	✓	×	✓	×	×	×	×	×
90	saucepan	×	×	×	×	×	×	×	×	×	×	×	×
91	screwdriver	✓	✓	✓	✓	✓	✓	×	✓	×	✓	×	×
92	scrub_brush	✓	✓	✓	✓	✓	✓	×	✓	✓	✓	×	×
93	snow_globe	✓	×	✓	×	×	×	✓	×	✓	×	×	×
94	teacup	✓	×	✓	×	×	✓	×	×	✓	✓	✓	×
95	tube_of_toothpaste	✓	✓	×	✓	✓	✓	✓	✓	✓	✓	×	✓
96	vegetable_peeler	✓	✓	✓	×	✓	✓	×	✓	✓	✓	×	×
97	water_glass	×	✓	✓	✓	×	×	✓	×	×	×	×	×
98	toy_figure	✓	✓	×	×	×	×	×	×	×	×	×	×
99	spirit_lamp	✓	×	✓	✓	×	✓	✓	✓	×	×	✓	×
100	zucchini	✓	×	✓	✓	✓	✓	×	×	✓	×	×	×