The Harness: One Agent Harness for Anything that Acts

Sihan Xu[1,6] **Zilin Wang**[1*] **Hui Shen**[1,6*] **Zhengjie Xu**[1*] **Weiyang Jin**[6*]
Yuwei Niu[2,6] Ye Li[1] Jiechen Zhang[3] Yinpei Dai[1] Yunhao Luo[1]
Wenhao Chai[4] Xuweiyi Chen[5] Ziwei Liu[5,6] Stella Yu[1]

[1]University of Michigan [2]Nanyang Technological University [3]EPFL [4]Princeton University
[5]University of Virginia [6]LMMs-Lab
[*]Equal contribution

Abstract: Vision-language-action (VLA) policies are skilled but narrow, while frontier models asked for every action are general but slow and unsafe. The Harness places a coding agent at the top of a three-layer control hierarchy: it writes programs over perception and motion skills, calls served VLAs as bounded skills, and writes guards that react at every control tick without a model call, above a fixed safety shield. Because language models, learned policies, and embodiments enter through configuration, the same harness also operates desktop computers and plays commercial games, with GUI action models in place of VLAs. On the full RoboDojo protocol, one low-cost language model raises the Simate-beta policy it calls from 33.95 to 39.59, at only 3.3 model calls and $0.006 per episode. Videos and more results can be found at https://harness.lmms-lab.com/

Keywords: Agentic robotics, Vision-language-action models, Code as policies

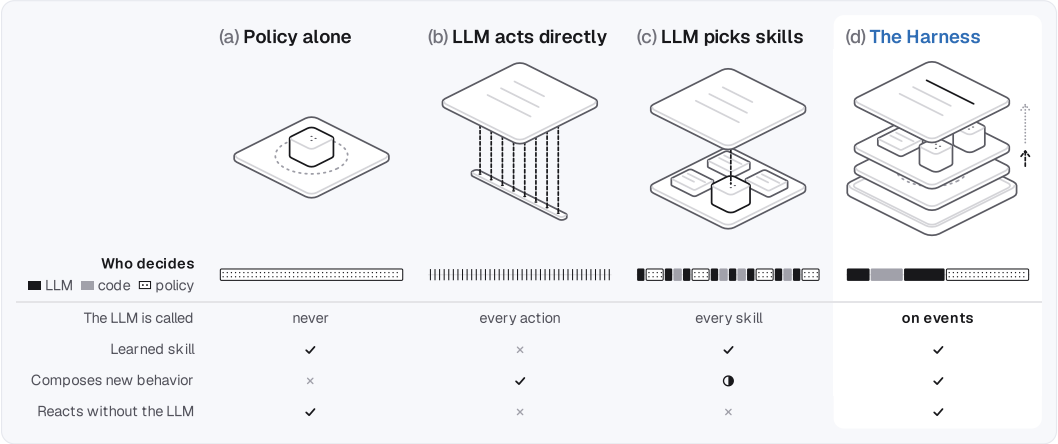


Figure 1: Four ways to build a robot policy. (a) A VLA alone; (b) a language model for every action; (c) a language model picking one skill per call; (d) The Harness, whose model writes the programs, skills and guards, and is called only on events. The strip shows who decides over one episode

1 Introduction

VLA models act quickly and skillfully within their training data [1, 2, 3, 4], while frontier language models asked for every action generalize but wait on a model call at every step [5]. On RoboDojo [6], a frontier model choosing end-effector targets scores 12.7 on precision tasks against 43.0 for the strongest trained policy, and its evaluation on physical robots was halted after unsafe actions [5]. Classical robots resolved this split with layered control [7, 8, 9], but engineers wrote every layer by hand; The Harness has a coding agent write them (Fig. 1d). Unlike agents that write programs without learned policies [10, 11, 12, 13] or orchestrate frozen VLAs as fixed primitives [14, 15, 16, 17], it calls any served VLA as one bounded skill among others and programs a reactive layer that runs without the model.

10th Conference on Robot Learning (CoRL 2026), Austin TX, USA.

Table 1: RoboDojo, full protocol (Score per dimension; average Score and SR, %). Baselines: leaderboard of 2026-10-14; The Harness is seed 0 with DeepSeek V4.1 Flash. Calls and cost per episode. †10 episodes per task. ‡Estimated by its authors.

Method	Type	Gen.	Prec.	Long	Mem.	Open	Score	SR	Calls	Cost ($)
VPP2	WAM	36.38	43.03	56.28	56.01	4.61	39.26	32.26	–	–
Simate-beta	VLA	35.09	34.35	57.84	33.33	9.12	33.95	27.96	–	–
$\pi_{0.5}$	VLA	13.38	12.40	23.54	5.89	1.98	11.44	6.93	–	–
RoboProbe, GPT-6-Astra	Agent	33.36	12.65	21.45	43.04	34.36	28.97	22.48	$\leq$100	–
RoboProbe, DeepSeek-Flash†	Agent	2.42	1.75	2.12	2.17	6.50	2.99	1.92	$\leq$100	–
PhysicalRSI	Agent + VLA	21.30	38.05	46.37	46.74	28.88	36.27	31.38	–	–
EmbodiedEvo	Agent	35.22	25.15	33.83	79.40	51.81	45.08	38.95	–	5.73‡
The Harness + Simate-beta	Agent + VLA	36.69	37.02	59.81	47.89	16.54	**39.59**	**33.54**	**3.3**	**0.006**
Δ vs. Simate-beta		+1.60	+2.67	+1.97	+14.56	+7.42	+5.64	+5.58		
Human teleoperation	–	*90.05*	*68.06*	*83.63*	*75.25*	*85.13*	*80.42*	*76.03*	–	–

2 Method

In The Harness, every skill runs as a *bounded execution* for at most T seconds, inside a workspace region, until a predicate holds, ending with an *event* (completed, timeout, guard triggered, policy stalled, command rejected).

High level. The language model observes the instructions and states, and write a Python program at the start, when a program ends, or when an event is unresolved. It never selects individual commands.

Low level. Programs compose analytic skills (SAM 3 segmentation [18], pointing, grasp and motion planning, behind tools such as `locate` and `pick`), functions the agent writes, and `run_policy`, which runs any VLA served through XPolicyLab [19]. From a capability description of the VLA, the program will choose among hands VLA the task, hands VLA over and finishes what is left, or keeps the task (Fig. 2).

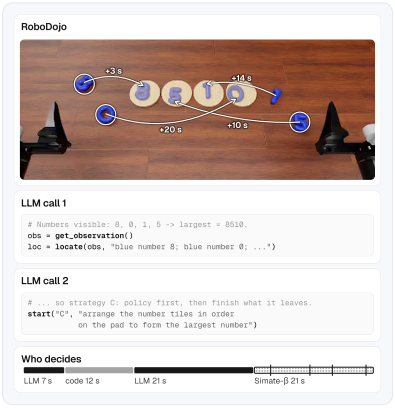


Figure 2: First locates the digits of `arrange_largest_number` and then hands the task to Simate-beta. Cost: $0.002.

Reactive layer and shield. Guards (`when`, `watch`) and controllers (`run_controller`) written by the model run at every 40 ms tick without model calls. An event reaches the model only if no handler resolves it. An underlying fixed shield rejects invalid operations i.e. out-of-limit joint actions, and never alters a command. Models, policies, and robots each enter through one configuration file.

3 Results on RoboDojo

Setup. RoboDojo [6] has 42 simulated bimanual tasks, 50 episodes each. The agent sees only the instruction, camera images, and robot state. Every model role uses DeepSeek V4.1 Flash, the learned skill is the public Simate-beta checkpoint, and **one frozen configuration** serves all tasks.

Main result. The Harness raises Simate-beta from 33.95 to 39.59, in every dimension and above every trained policy on the leaderboard (Table 1). The largest gains are where the policy cannot read the instruction: `press_by_number` (0.67 to 88) and `solve_equation` (0 to 46). An episode averages 3.3 calls and $0.006, and the full 2,100-episode evaluation costs about $13 in model calls.

Ablations. On eight comparison tasks, Simate-beta alone scores 41.6 and The Harness 46.0 at $0.006 per episode. With the same model, choosing one skill per call scores 28.2 at $0.012, and programs without the policy score 10.7 at $0.066. Changing one file, the stronger Grok 4.7 scores 43.0 at $0.110, and GalaxeaVLA G0.5 rises from 25.0 to 32.3 under the harness.

Reaction and safety. Guards react without waiting for the model. In a scripted test, the open gripper descends onto an object with no model in the loop. A guard that stops the arm once it stops moving down within two control ticks (80 ms) of contact; without the guard, the arm keeps pushing for a median 0.56 s until the motion skill reports. Asking the model instead would take a median 34.2 s per decision. Tested on 50,049 generated commands, the shield let no invalid command through.

References

[1] A. Brohan, N. Brown, J. Carbajal, et al. RT-2: Vision-language-action models transfer web knowledge to robotic control. In *Conference on Robot Learning (CoRL)*, 2023.

[2] M. J. Kim, K. Pertsch, S. Karamcheti, et al. OpenVLA: An open-source vision-language-action model. In *Conference on Robot Learning (CoRL)*, 2024.

[3] K. Black, N. Brown, D. Driess, et al. π_0: A vision-language-action flow model for general robot control. In *Robotics: Science and Systems (RSS)*, 2025.

[4] Physical Intelligence et al. $\pi_{0.5}$: A vision-language-action model with open-world generalization. *arXiv preprint arXiv:2504.16054*, 2025.

[5] W. Zhang, K. Wang, Y. Ouyang, et al. An unexpected robot policy: Early evaluations of GPT-6 Astra on RoboDojo and beyond. *arXiv preprint arXiv:2609.24170*, 2026.

[6] T. Chen, Y. Chen, Z. Li, et al. RoboDojo: A unified sim-and-real benchmark for comprehensive evaluation of generalist robot manipulation policies. *arXiv preprint arXiv:2607.04434*, 2026.

[7] R. J. Firby. *Adaptive Execution in Complex Dynamic Worlds*. PhD thesis, Yale University, 1989.

[8] R. P. Bonasso, R. J. Firby, E. Gat, D. Kortenkamp, D. P. Miller, and M. G. Slack. Experiences with an architecture for intelligent, reactive agents. *Journal of Experimental and Theoretical Artificial Intelligence*, 9(2–3):237–256, 1997.

[9] E. Gat. On three-layer architectures. In D. Kortenkamp, R. P. Bonasso, and R. Murphy, editors, *Artificial Intelligence and Mobile Robots*, pages 195–210. MIT Press, 1998.

[10] J. Liang, W. Huang, F. Xia, et al. Code as policies: Language model programs for embodied control. In *IEEE International Conference on Robotics and Automation (ICRA)*, pages 9493–9500, 2023.

[11] L. Fu, J. Yu, K. El-Refai, et al. CaP-X: A framework for benchmarking and improving coding agents for robot manipulation. *arXiv preprint arXiv:2603.22435*, 2026.

[12] R. Lu, Y. Wu, E. Kou, et al. ASPIRE: Agentic skills discovery for robotics. *arXiv preprint arXiv:2607.00272*, 2026.

[13] M. Jia, Y. Lin, X. Zhang, Z. Zhang, X. Liu, and M. Jiang. Agent as policy for robotic manipulation. *arXiv preprint arXiv:2609.12541*, 2026.

[14] Y. Zhang, H. Zhang, F. Gao, et al. Harness VLA: Steering frozen VLAs into reliable manipulation primitives via memory-guided agents. *arXiv preprint arXiv:2607.08448*, 2026.

[15] K. Chen, S. Xie, L. Fu, et al. GaP: A graph-as-policy multi-agent self-learning harness for variational automation tasks. *arXiv preprint arXiv:2607.05369*, 2026.

[16] L. Galanti, D. Shah, and T. Dao. Addressing the orchestration gap in generalist robots via physical agency. *arXiv preprint arXiv:2607.21725*, 2026.

[17] R. Li, Y. Zhou, Y. Zhu, et al. RoboClaw: An agentic framework for scalable long-horizon robotic tasks. *arXiv preprint arXiv:2603.11558*, 2026.

[18] N. Carion et al. SAM 3: Segment anything with concepts. *arXiv preprint arXiv:2511.16719*, 2025.

[19] XPolicyLab Community, T. Chen, Y. Chen, et al. XPolicyLab: A unified standard and open ecosystem for robot policy evaluation and deployment. *arXiv preprint arXiv:2608.09892*, 2026.