CORL 2026 · AUSTIN, TX

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]

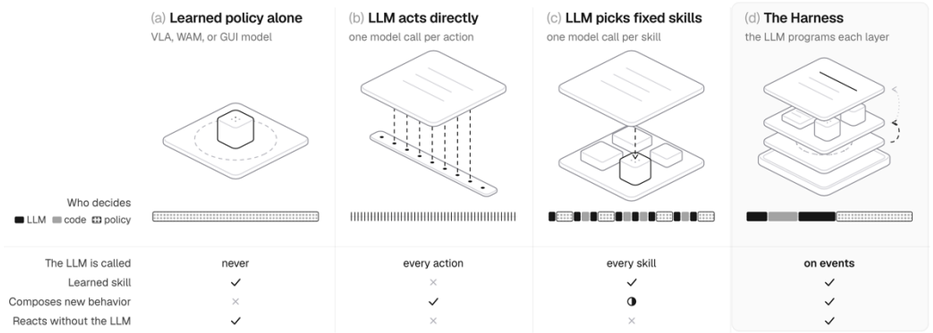
[1]University of Michigan [2]Nanyang Technological University [3]EPFL [4]Princeton University [5]University of Virginia [6]LMMs-Lab [*]Equal contribution

harness.lmms-lab.com

Videos, the platform and more results

Why The Harness?

VLAs and GUI action models are skilled and fast but narrow. A frontier model asked for every action is general but slow, costly and unsafe. In **The Harness (d)** the model writes every layer and is called only on events.

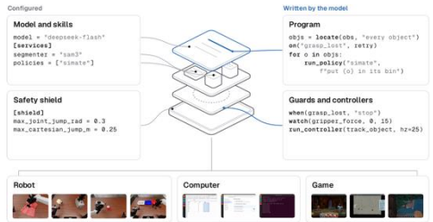


	(a) Learned policy alone	(b) LLM acts directly	(c) LLM picks fixed skills	(d) The Harness
The LLM is called	never	every action	every skill	**on events**
Learned skill	✓	×	✓	✓
Composes new behavior	×	✓	◑	✓
Reacts without the LLM	✓	×	×	✓

The Harness

Code first, then a learned policy, then LLM

Classical robots used three hand-written layers: deliberate, sequence, react. Here a coding agent writes all three, calls learned policies as skills, and sits above a fixed shield.



Left: what enters through configuration. Right: what the model writes. The same stack runs desktops (OSWorld) and games (GameScientist); those evaluations are in progress.

How each decision is made

High level. At the start, when a program ends, or when an event is unresolved, the model reads the instruction, images and traces and writes a Python program.

Low level. Programs compose bounded skills: SAM 3, pointing, grasp and motion planning, agent-written helpers, and run_policy for any served VLA. From a VLA's capability description, the first program hands the task over, keeps it, or does both.

Reactive layer. Guards (when, watch) and controllers (run_controller) written by the model run at every 40 ms tick, with no model call.

Shield. A fixed, non-learned check outside the agent's process rejects invalid commands and never alters one.

Escalation. An event goes to the program's handlers first and reaches the model only if none resolves it.

Fewer calls, lower cost at $0.002



Call 1 reads the digits (largest: 8510) and writes a program that only looks; call 2 hands the task to Simate-β, which places the tiles.

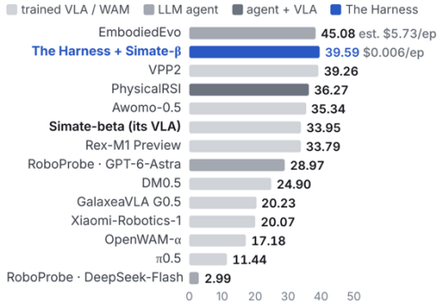
Reacting without the model

6 µs to evaluate a guard	**1.8 ms** per controller frame; a tick is 40 ms	**34.2 s** one model decision (median)
2 ticks to stop at contact; 0.56 s later without a guard	**50,049** test commands; 0 invalid passed	**4 / 4** injected faults stopped the robot

The shield only rejects: one that also clipped velocity cut G0.5 from 46.4 to 17.1; rejecting only, 47.2.

RoboDojo Result

39.59	3.3	$0.006
Score; 33.95 for Simate-beta	model calls per episode	per episode; ≈$13 in total



Above every trained policy, and above Simate-beta in every dimension: Gen. **+1.60**, Prec. **+2.67**, Long **+1.97**, Mem. **+14.56**, Open **+7.42**; SR 27.96 → 33.54.

Learning from past episodes

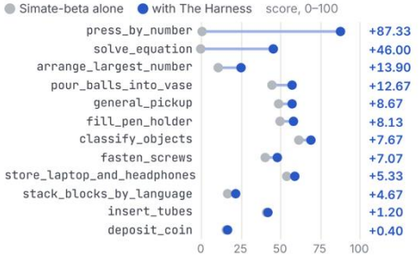
A weak harness without our written experience regrows it as a validated library (18 candidates, 1 note kept), tested on 20 unseen layouts.



Kept note: call run_policy early for insertions the perception skills cannot localize.

Where the gains come from

12 tasks higher, 30 equal, none lower; the largest gains are rules the VLA cannot read.



Which layer does the work?

Same model, shield and budget on 8 comparison tasks.



† RoboProbe, DeepSeek-Flash, full protocol.

Any language model, any VLA

Each swap edits one file; the harness is unchanged.

