vfm-eval.github.io

better evals for better policy



Tesla Optimus Demo, May 2025

Kamal, ex. Tesla Optimus

https://x.com/kamalgupta09

TALK STRUCTURE

Three ways to evaluate. Three ways to improve.

FIRST · MEASURE

Classical simulation

World-model simulation

Real-world trials

THEN · LEARN

Inverse dynamics

Policy conditioning / RL

Better world models / simulators

Evaluate → retain trajectories → improve → evaluate again

Classical simulation gives us controlled experiments

Strengths

Repeatable resets and initial states

Explicit state and success checks

Systematic task and scene variation

Limitations

Contact and control mismatch

Asset and physics engineering

Benchmark-specific shortcuts

Examples: RoboCasa365 · SIMPLER · LIBERO · Colosseum

Best for broad regression tests and controlled ablations.

[1] RoboCasa365 · [2] SIMPLER · [3] LIBERO · [4] Colosseum

RoboCasa makes coverage a controlled variable

365

everyday tasks

2,500

kitchen environments

50

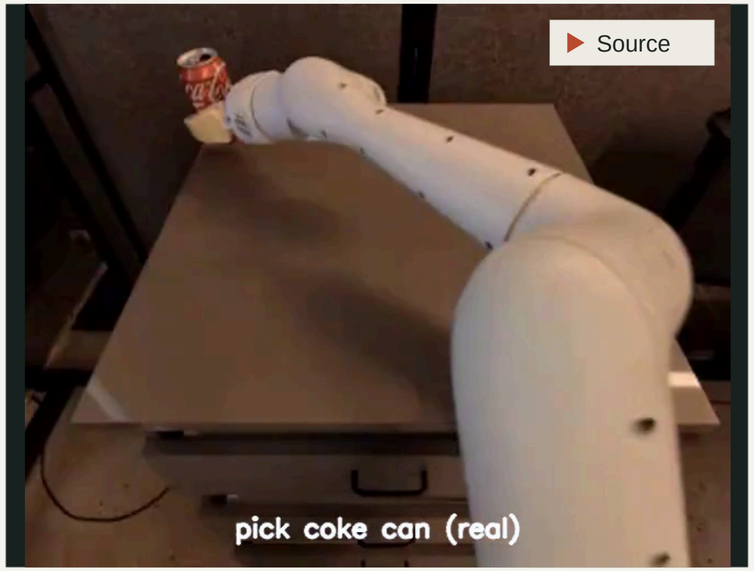
target benchmark tasks



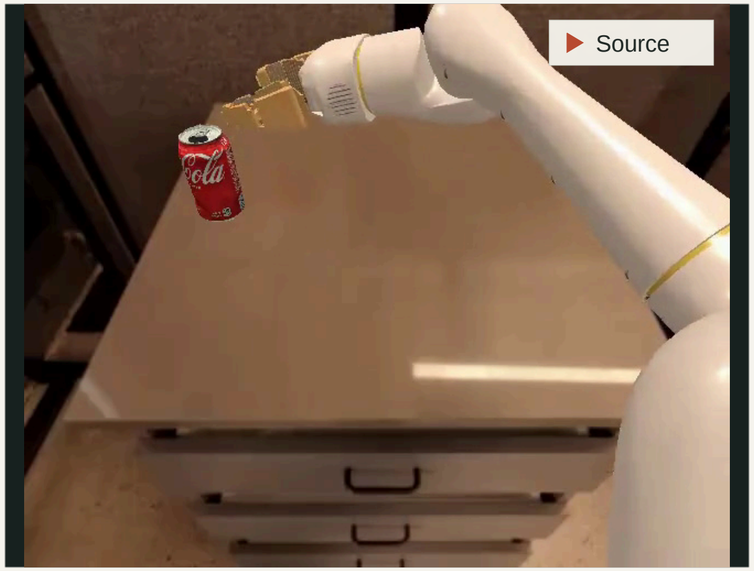
RoboCasa environment gallery

[1] RoboCasa365 · [2] RoboCasa release

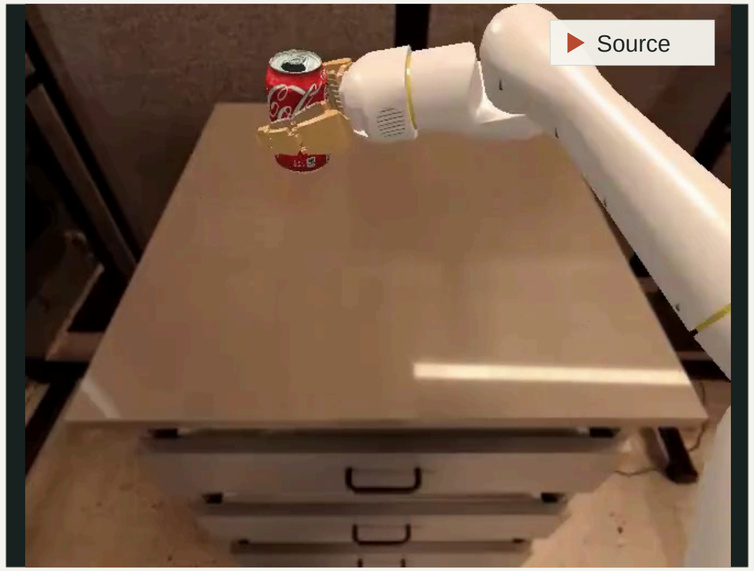
Match control before trusting a sim score



REAL



SIM · before system identification

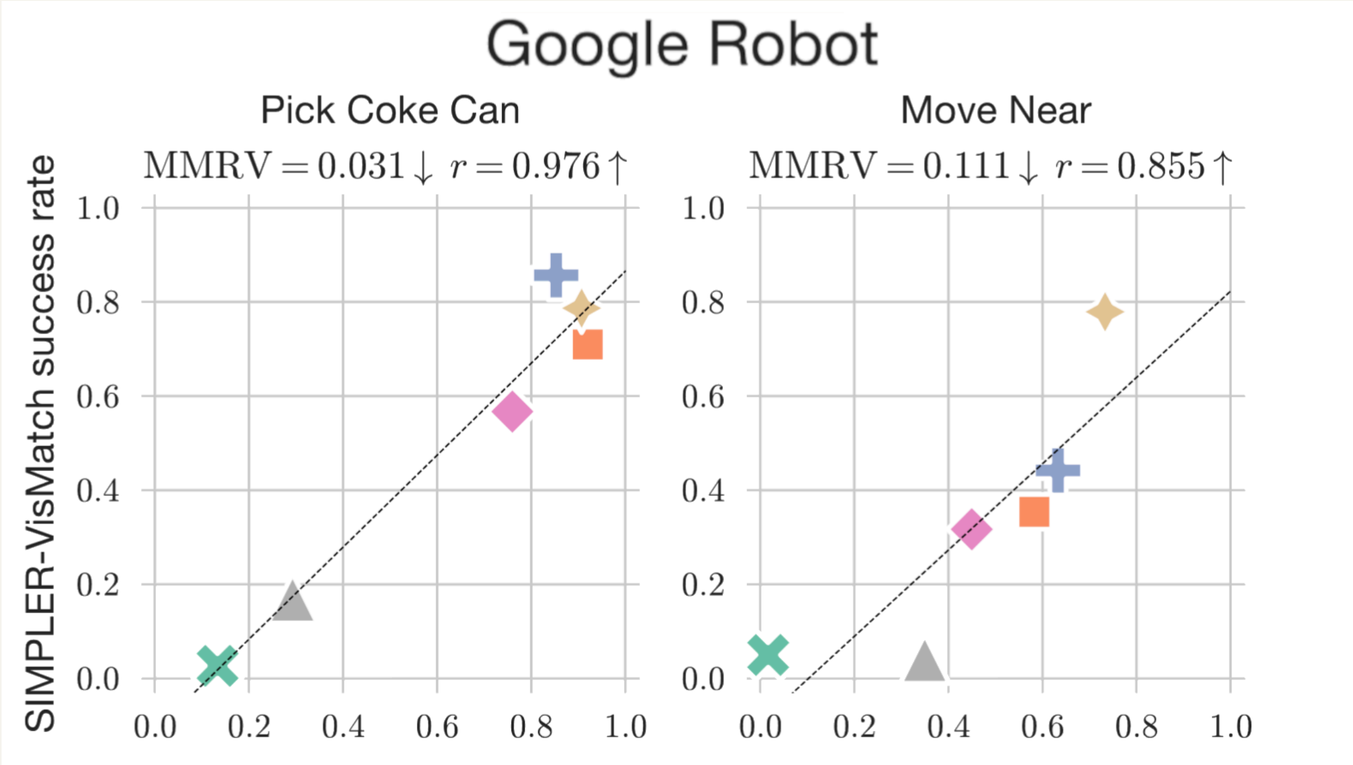


SIM · after system identification

The same action must have a comparable physical effect.

[1] SIMPLER

Ranking agreement is the sim-to-real test



SIMPLER · two Google Robot tasks · each marker is a policy / checkpoint

x: real success **y:** sim success

MMRV ↓ penalizes consequential rank reversals.

[1] SIMPLER

Generative tools can help build a classical simulator



Astra · Blender scene authoring



Cosmos-Transfer2.5 · generated appearance

Geometry, rendering, dynamics, and the task checker are separate components.

[1] Astra Blender · [2] Cosmos Cookbook

World-model simulation learns the transitions

Strengths

Start from recorded observations

Model visual complexity from data

Compare counterfactual action sequences

Limitations

Prediction drift over long horizons

Unfamiliar actions, objects and contact

Optimistic or incorrect outcomes

Examples: 1XWM · WorldEval · Ctrl-World · GigaWorld-1

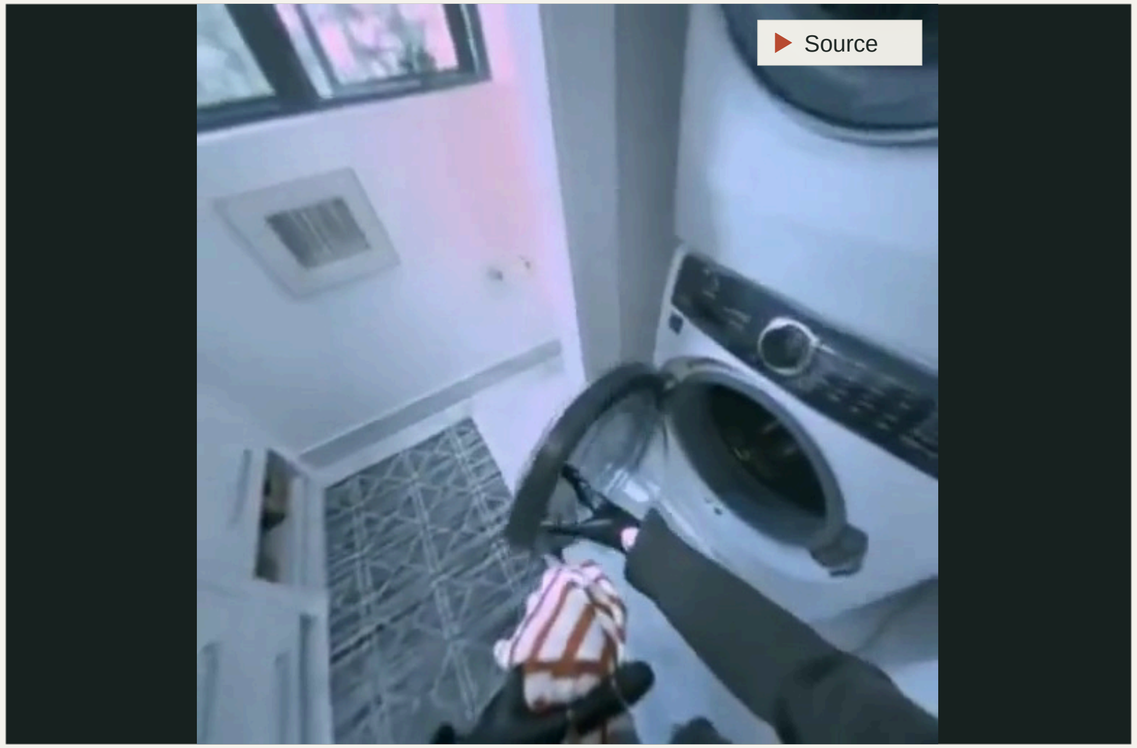
Best for screening candidates within a validated data distribution.

[1] 1X 2025 · [2] WorldEval · [3] Ctrl-World · [4] GigaWorld-1

1X predicts consequences of specific actions



REAL · recorded robot actions

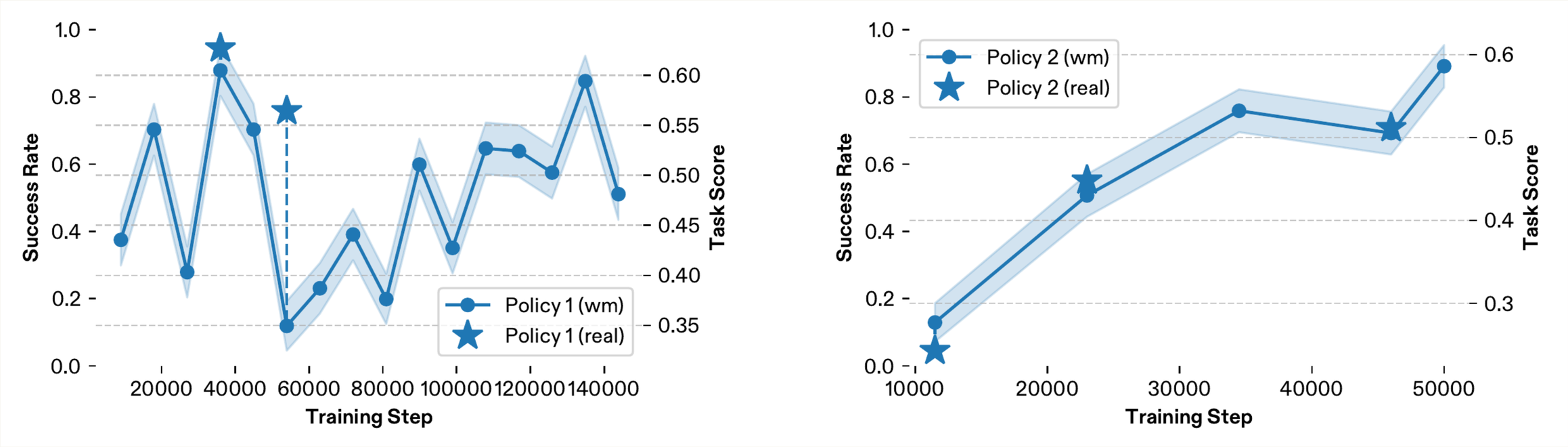


GENERATED · conditioned on those actions

Same starting observation. Action-sensitive future. Learned outcome score.

[1] 1X 2025 · [2] 1X report

The practical payoff is checkpoint selection



1X report · Figure 7 · lines: world model · stars: real trials

Screen many checkpoints

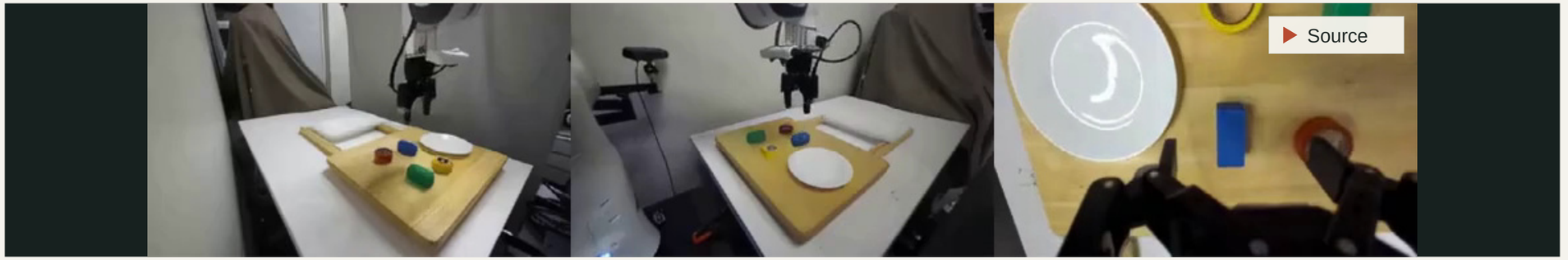
Spend real trials on promising candidates.

Read both axes

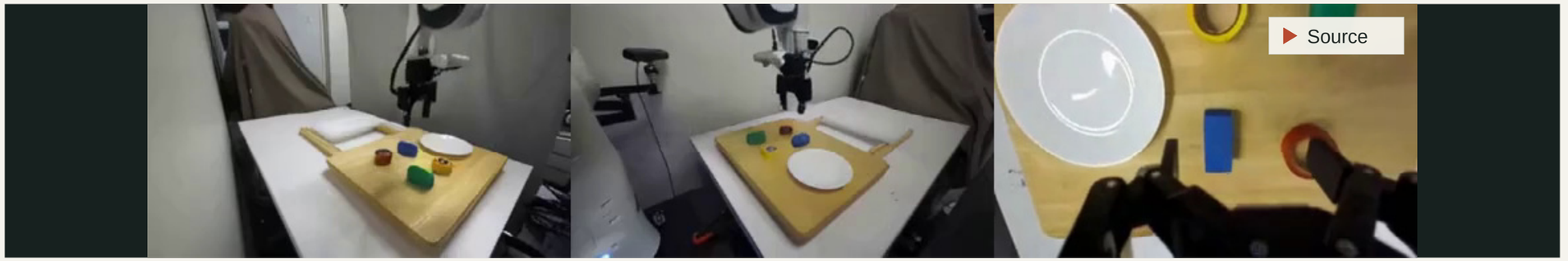
WM success rate and real task score differ.

[1] 1X report

Ctrl-World keeps the policy in the loop



REAL · multi-view execution



GENERATED · multi-view rollout

Observation → policy → action chunk → world model → next observation

Tesla replays historical failures with a new policy

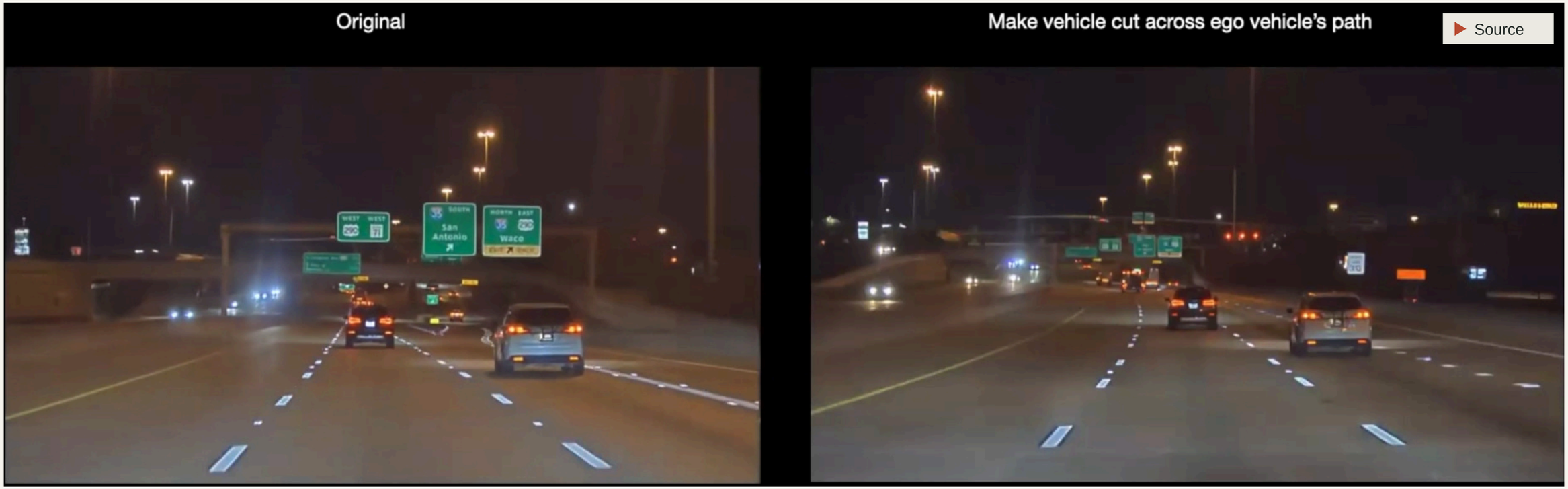


Tesla driving example · historical issue / new-policy simulation

Replay a known failure before accepting the next policy.

[1] Tesla · CVPR WAD 2026

Perturb a recorded scene to test a new failure



Tesla driving example · original scene / synthetic cut-in

Use a collected case to build a harder test.

Real-world trials measure the actual system

Strengths

Actual sensors, contact and hardware

End-to-end behavior and interventions

Reference evidence for simulator validity

Limitations

Slow resets and limited robot time

Environment drift and wear

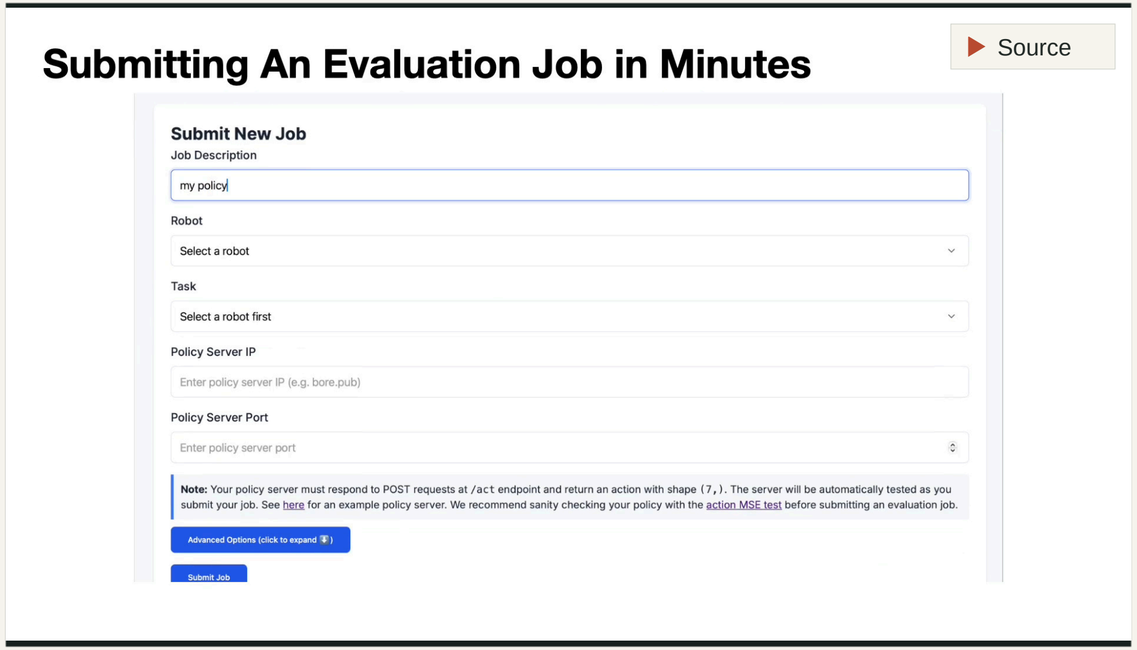
Many trials needed for small gains

Examples: RoboArena · AutoEval · Gemini Robotics · Robocurve

Best for validating transfer and making the final performance claim.

[1] RoboArena · [2] AutoEval · [3] Gemini Robotics 1.5 · [4] Robocurve

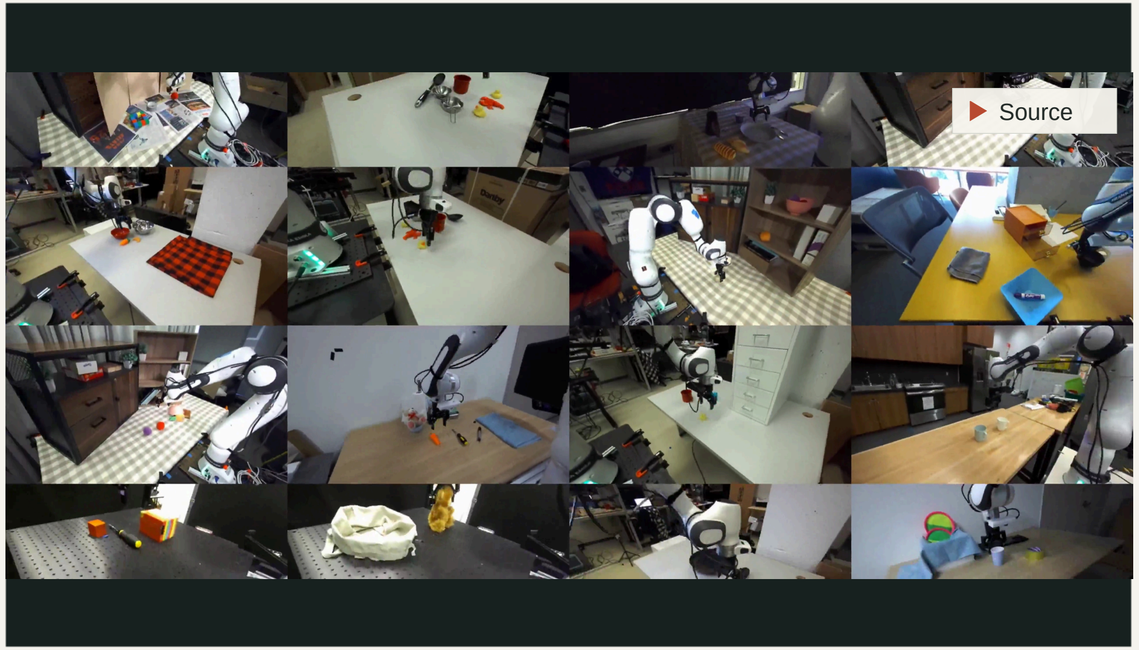
Real robots keep the proxies accountable



AutoEval · autonomous evaluation demo (2025)

Repeatability

Automate the physical experiment.



RoboArena · distributed paired evaluation

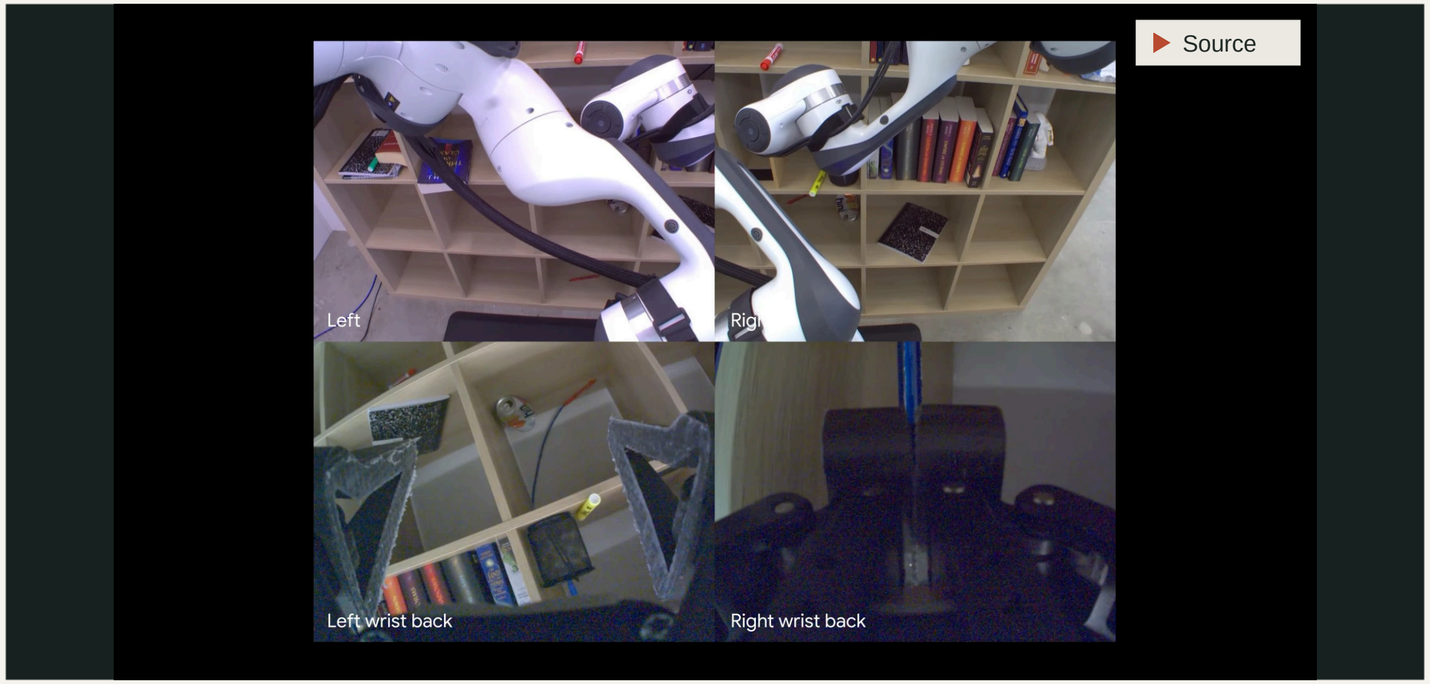
Diversity

Compare blindly across environments.

[1] AutoEval · [2] RoboArena

SCORING · ACROSS ALL THREE SETTINGS

The judge is another model to evaluate



Gemini Robotics-ER 1.6 · multi-view success detection

Observe across views

Resolve occlusion
before declaring success.

Audit the reward

Near misses and
false positives matter.

Validate the outcome judge in simulation and in reality.

[1] Gemini Robotics-ER

EVALUATION · COMPARISON

The three approaches are complementary

CRITERION	CLASSICAL SIM	WORLD-MODEL SIM	REAL WORLD
Fidelity	~ Explicit approximations	~ Learned approximations	+ Actual target system
Repeatability	+ State resets	+ Seeds + fixed starts	– Resets + environment drift
Coverage	+ Authored variations	+ Data-supported variations	– Physical setups available
Cost	+ Physics/render compute	~ Inference compute	– Robot + human time
Scoring	+ State predicates / judges	~ Predicted states / judges	~ Sensors / judges / audits

Validate the proxy on the policy decision that matters.

+ Typically stronger ~ Depends on setup – Typically weaker Qualitative comparison

FROM EVALUATION TO IMPROVEMENT

Keep the trajectory, not just the score

observation → executed action → next observation

Recover actions	Train inverse dynamics
Improve behavior	Condition or train the policy
Improve the environment model	Fit the world model or simulator

Also retain outcomes, timing, initial conditions, and human corrections.

Inverse dynamics learns the action behind a transition

INPUT

Observation now + observation later

→

PREDICT

Action sequence

01

Collect

Aligned robot transitions

→

02

Supervise

Use executed action labels

→

03

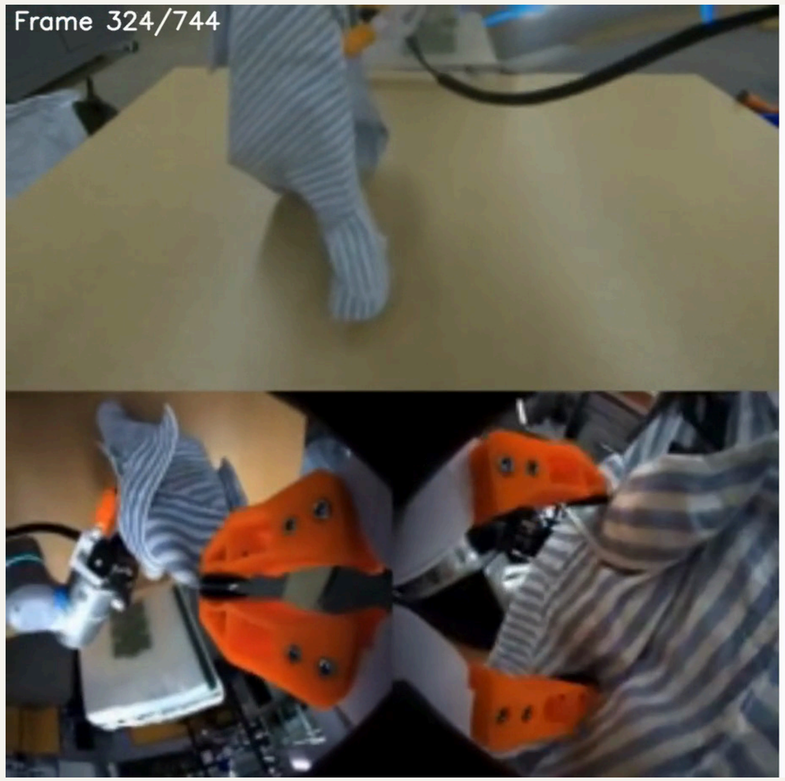
Apply

Label videos or track futures

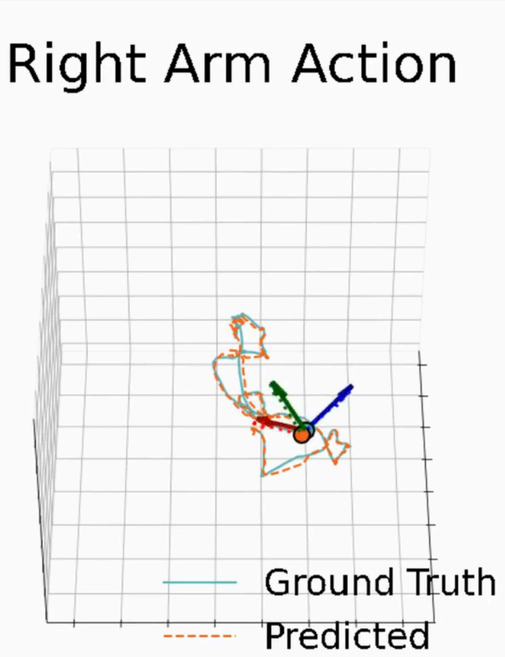
A failed task can still provide a valid action–transition example.

Rhoda makes inverse dynamics visible

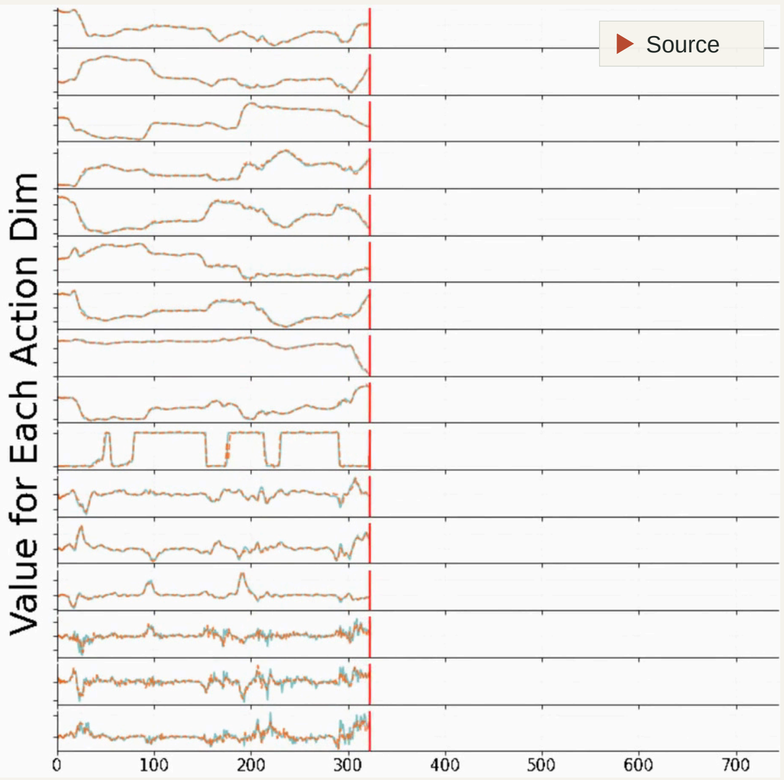
INPUT · robot video



OUTPUT · recovered motion



CHECK · predicted vs. recorded



Rhoda AI · video-to-action translation · source demo at 1×

Proposed reuse: train the translator on action-labeled eval logs.

[1] Rhoda AI · Direct Video-Action

Improve policies with quality labels and rewards

Quality conditioning

Label how well and how quickly each attempt was performed.

Train with the label.
Request desirable behavior.

Example: $\pi_{0.7}$

Reward and advantage

Use outcomes to estimate which actions helped.

Update the policy toward better expected outcomes.

Example: $\pi^*_{0.6}$ / RECAP

Retain imperfect attempts without treating every action as expert behavior.

[1] $\pi_{0.7}$ conditioning · [2] RECAP paper

RECAP turns experience into a policy update

Offline RL pretraining → Task demonstration fine-tuning

01

Collect

Rollouts + corrections

→

02

Estimate value

Outcome + time to completion

→

03

Assign advantage

Which actions helped?

→

04

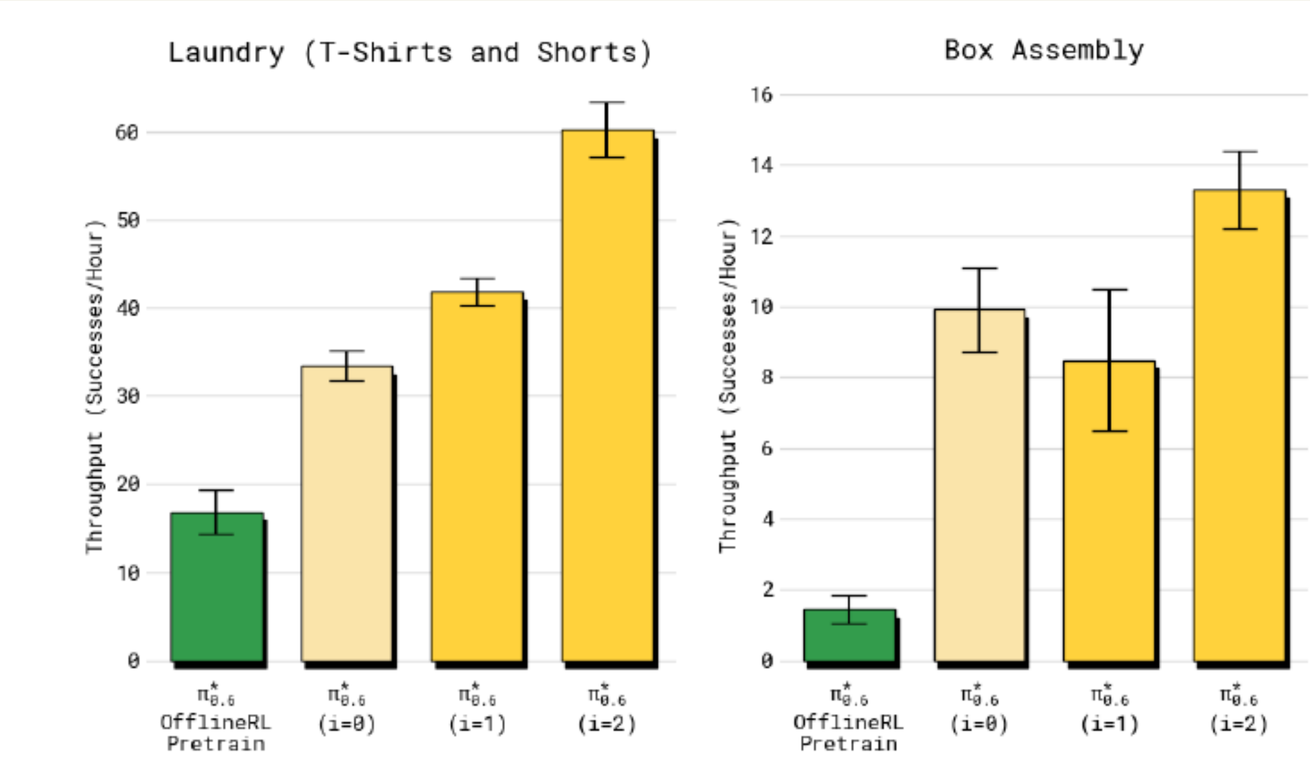
Update policy

Condition on improvement

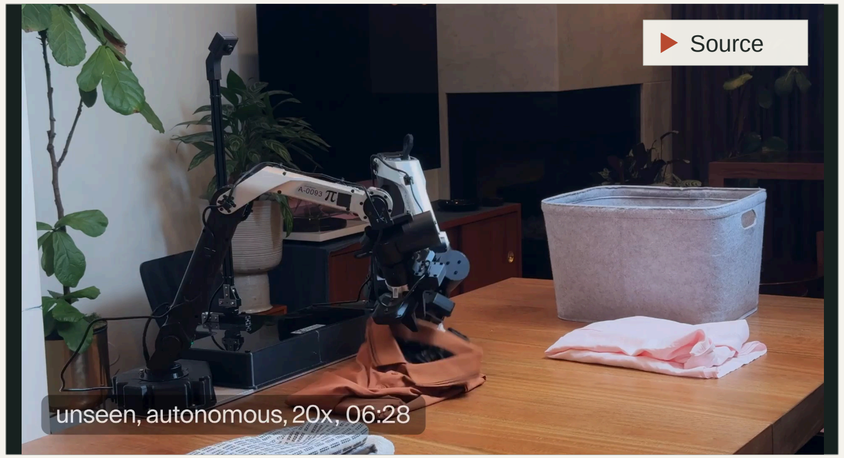
Run the updated policy. Collect a new batch. Repeat.

[1] RECAP paper

Hill climbing includes steps that do not improve



RECAP paper · Figure 9 · original reported iterations



$\pi^{\star}_{0.6}$ · illustrative laundry execution

Box throughput first dips, then rises.

Reevaluate every update.

[1] RECAP paper · [2] RECAP media

The same transitions can improve forward prediction

INPUT

Observation now
+ executed actions

→

PREDICT

Future observations

Train on discrepancies	Correct wrong contacts, occlusions and outcomes
Add missing coverage	Use the states the current policy actually visits

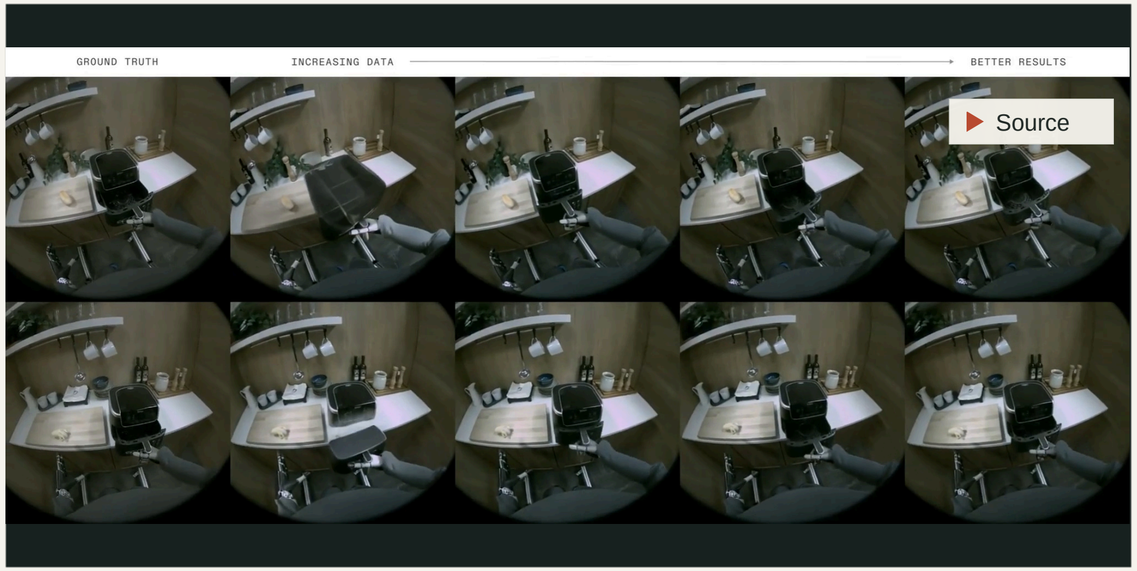
Inverse dynamics predicts actions. Forward dynamics predicts what follows them.

Real failures teach the world model how failure looks



Failure example · real left / generated right

Recorded More interaction data →



Separate example · increasing interaction data

1X reports that autonomous failure data reduces optimistic predictions.

[1] 1X report · [2] 1X 2025

Real rollouts can refine classical simulation too

01

Evaluate in real

Record the trajectory

→

02

Compare

Find sim–real discrepancies

→

03

Adapt the sim

Update parameter distributions

→

04

Train and recheck

Measure physical improvement

SimOpt	Fit randomization from real experience
AdaptSim	Adapt simulation for the target task

A better training simulator still needs validation as an evaluator.

[1] SimOpt · [2] AdaptSim

Protect the evaluation while learning from failures

Training stream	Failures, corrections, demonstrations → updates
Development suite	Regression tests → model selection
Sealed audit set	Unseen episodes + fresh real trials → final claim

Keep data used for learning or selection out of the held-out score.

MEASURE, IMPROVE, MEASURE AGAIN

Great. Can we do it again?

01

Evaluate

Classical / learned / real

→

02

Retain

Trajectories + outcomes

→

03

Update

IDM, policy, or simulator

→

04

Retest

Fresh held-out cases

→

05

Choose

Promote and iterate

Make “again” boring.

Choose a benchmark for the capability claim

BENCHMARK	PRIMARY LENS	WHAT STILL NEEDS VALIDATION
LIBERO	Spatial / object / goal transfer	Generalization outside its simulator
RoboCasa365	Household skills + task and scene scale	Physical transfer to the target robot
RoboTwin 2.0	Bimanual skills + domain randomization	Real setup and embodiment match
BEHAVIOR-1K	Long household activities + interaction	Full-system deployment reliability
Colosseum	Systematic perturbation robustness	Coverage of your actual shifts

[1] RoboCasa365 · [2] Colosseum · [3] LIBERO · [4] RoboTwin 2.0 · [5] BEHAVIOR-1K

“Unseen” has to name a split



RoboCasa · composite coffee task

18 atomic

Basic interaction skills

16 composite-seen

Present in pretraining

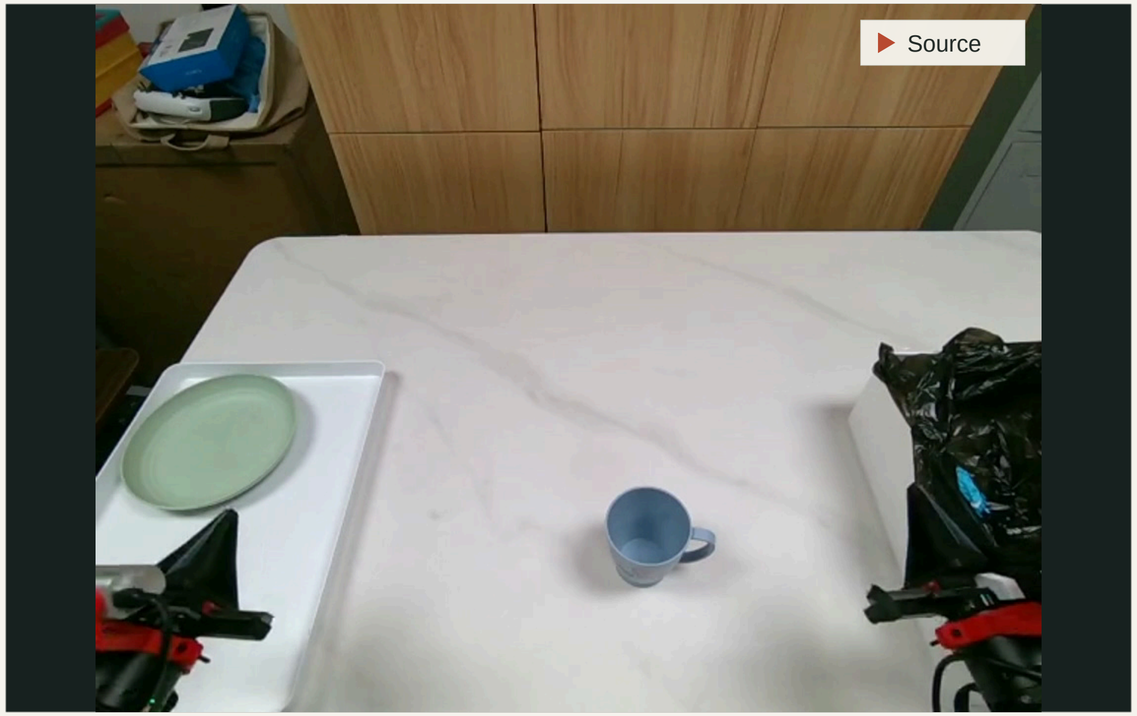
16 composite-unseen

Absent from pretraining

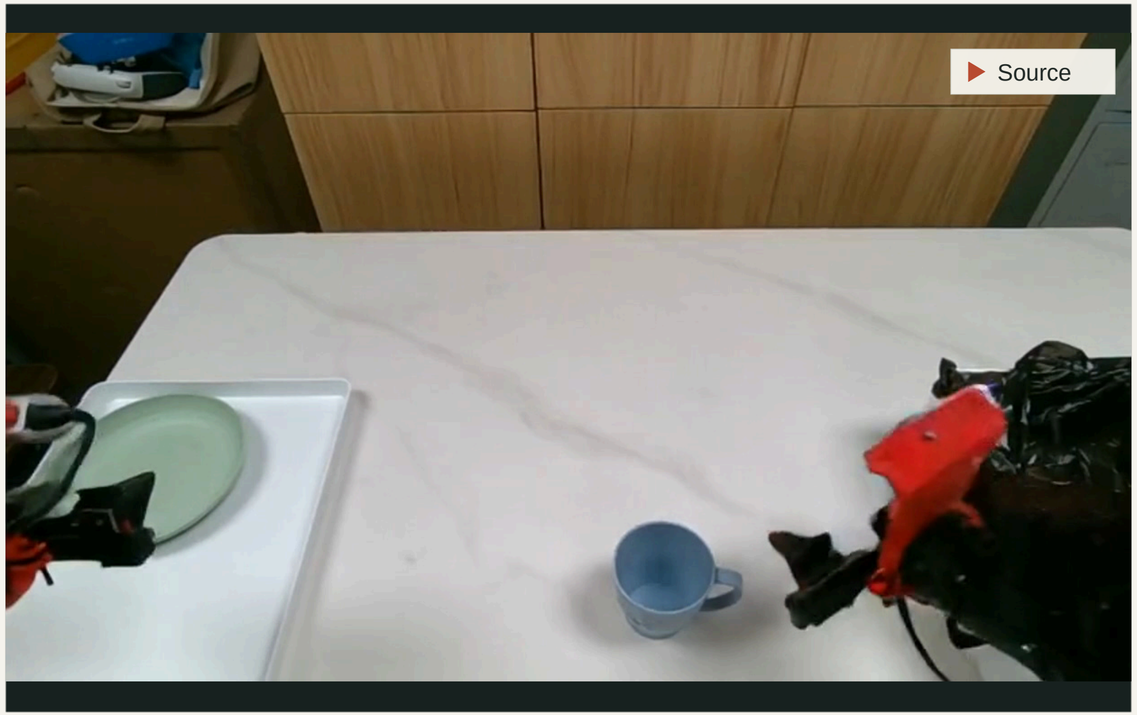
Target-task demonstrations may still be used for fine-tuning.

[1] RoboCasa365

WorldEval makes failure agreement visible



REAL · blue-mug failure



GENERATED · corresponding failure

A useful evaluator must preserve mistakes as well as successes.

[1] WorldEval · [2] WorldEval media

Action-conditioned simulation extends to Optimus



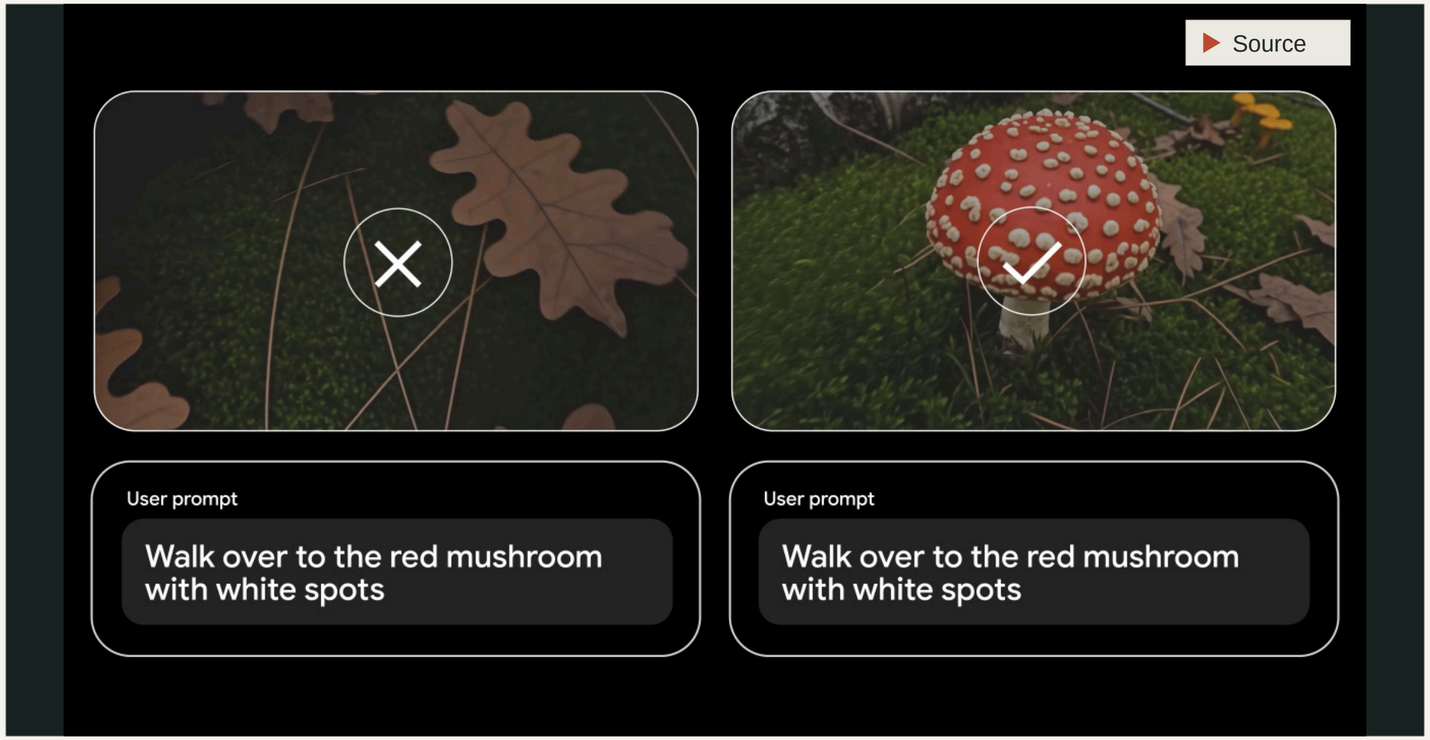
Recorded reference + generated straight / left / right



Generated drawer opening

Different actions should produce different consequences.

Genie + SIMA turn generated worlds into experience



SIMA 2 · improvement in a Genie 3 environment

Generate a world

Act toward a task.

Score with Gemini

Train on new experience.

Evaluate again

Test generalization.

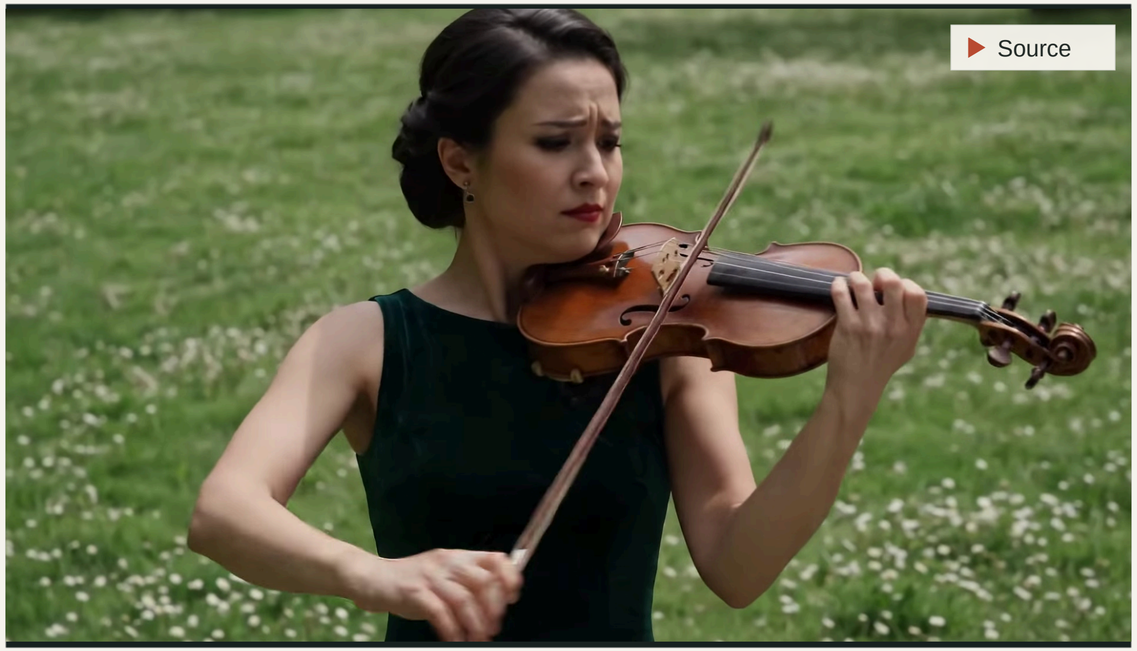
Demonstrated in virtual worlds. Physical robot transfer remains a separate test.

[1] Genie 3 · [2] SIMA 2 · [3] SIMA 2 report

Omni makes visual variation easier to propose



GEMINI OMNI · input video



GEMINI OMNI · edited scene

Candidate stress-test data; robot-policy validity still needs proof.

[1] Gemini Omni

Gemini Robotics separates generalization axes

230

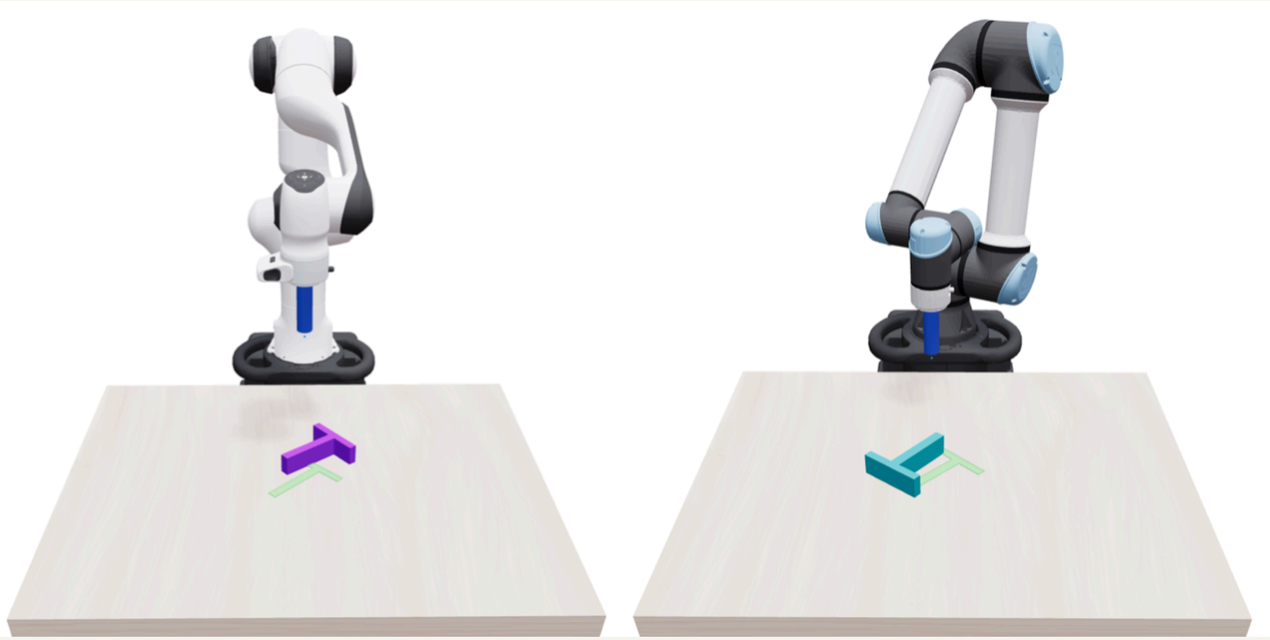
tasks in the GR 1.5 benchmark

Progress score
and binary success
answer different questions.

Visual	Backgrounds, lighting, distractors
Instruction	Language and paraphrases
Action	New motions and initial states
Task	New task in a new environment

[1] Gemini Robotics 1.5

Fable can hill-climb controller code in simulation



Agentic Push-T · simulated Franka and UR5e · original Figure 3

Write → run → diagnose → revise

The LLM changes the program.
The program controls the robot.

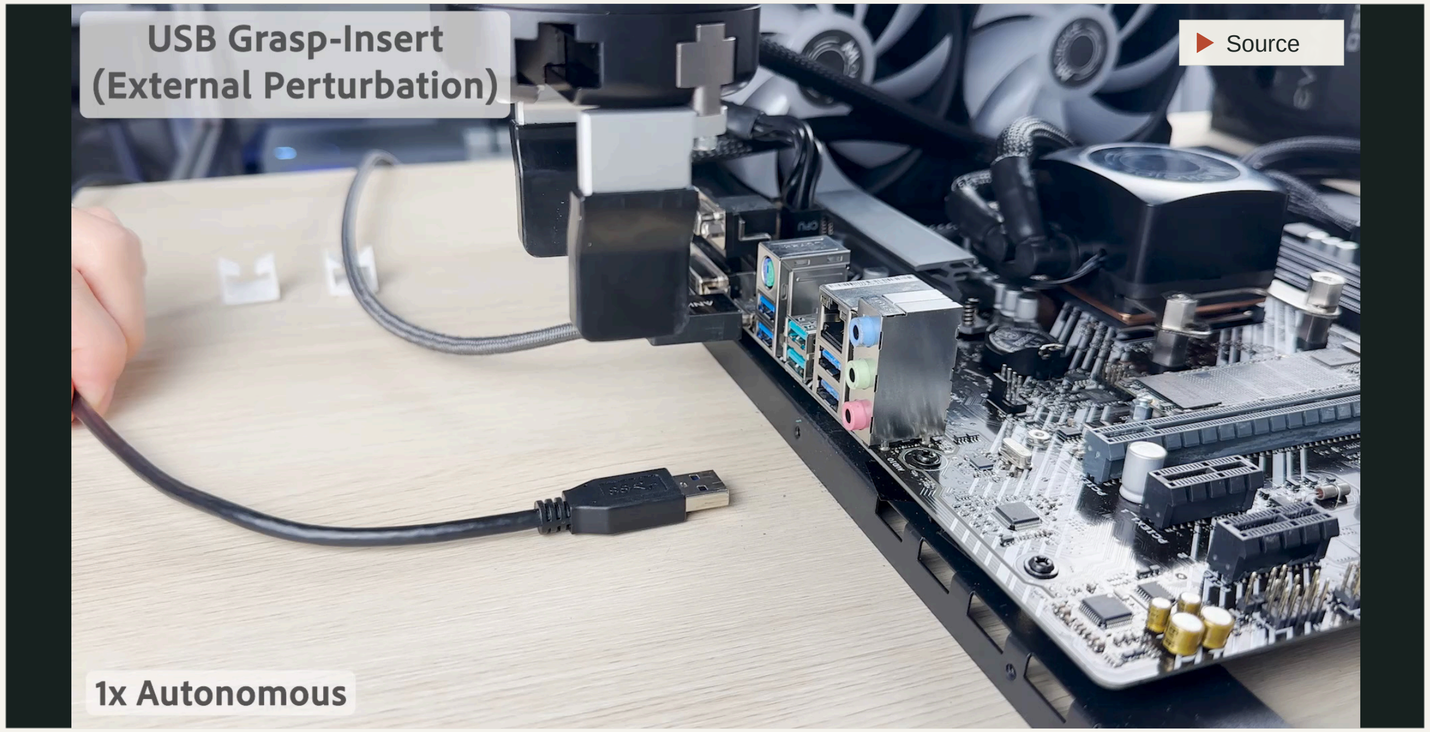
Check what the actor can see

Privileged state and RGB
are different experiments.

Reported simulated transfer is a step toward a real-robot test.

[1] Agentic Push-T

HIL-SERL shows the loop beyond large VLAs



HIL-SERL · USB insertion under perturbation

Reward classifier

Task outcome signal

Human intervention

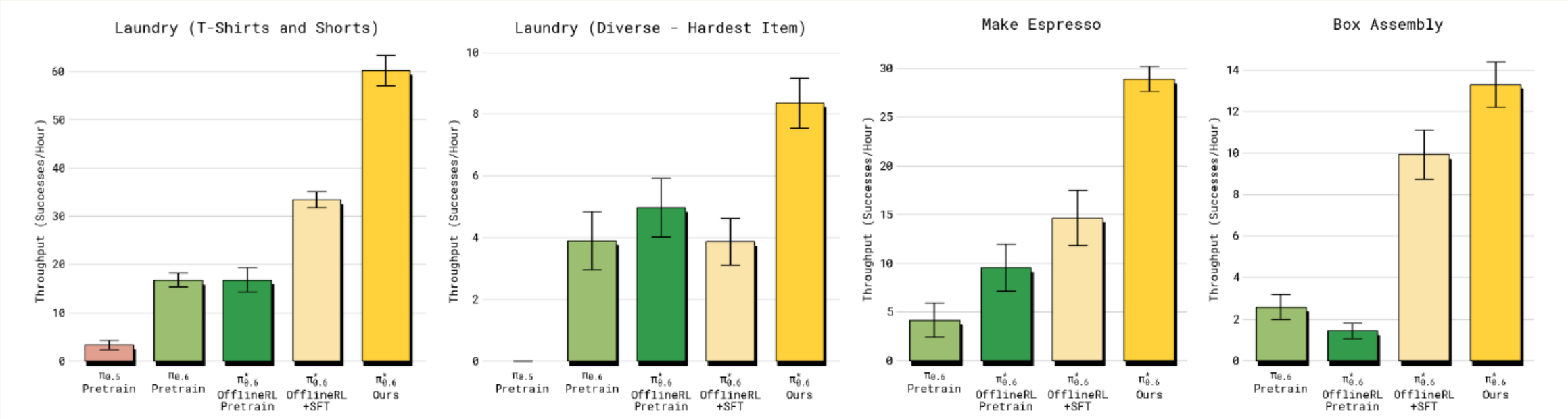
Recovery and exploration

Real-world RL

Improved success and cycle time

[1] HIL-SERL

Reward feedback improves successful work per hour



RECAP paper · Figure 7 · standard-error bars

Bars: $\pi_{0.6}$ → offline RL → + SFT → final; $\pi_{0.5}$ baseline where shown

[1] RECAP paper

Small gains can need many real trials

60 / 100 Policy A successes

vs.

70 / 100 Policy B successes

Observed gain: **10 percentage points**

Approximate 95% interval for B – A: **–3 to +23 points**

Illustrative independent binomial example · not a reported robot result

A practical proxy-validation experiment

Hold out	Policy families × objects × scenes × dates
Pair	Same candidates and comparable initial states
Measure	Rank violation, false promotion, calibration
Slice	Horizon, novelty, task, and failure category
Recheck	Whenever the policy or evaluator shifts

Three useful targets for evaluation data

Inverse dynamics	Which actions connect these observations?
Policy conditioning / RL	Which behavior should we prefer?
World model / simulator	What did our environment model get wrong?

One trajectory can support more than one learning target.