EVALUATING TEST-TIME SCALING IN ROBOT POLICIES AND WORLD MODELS

Anonymous authors
Paper under double-blind review

ABSTRACT

Vision-Language-Action (VLA) models and World Models (WMs) are becoming increasingly capable tools for robot decision-making, enabling action generation and prediction of future outcomes. These capabilities make sampling additional actions and their imagined futures a natural approach to scaling computation at test time. We introduce a diagnostic benchmark of sample-based test-time scaling that asks when additional samples improve action selection, and how well WM imagined futures can help with selecting the right action. Our evaluation spans VLAs, integrated World-Action Models (WAMs), VLAs augmented with action-conditioned WMs, and GPT-6 Astra with a waypoint scaffold. We organize the core metrics into three dimensions: sampling potential, measured by what is the best achievable gain from sampled actions assuming a perfect oracle selector is available (oracle headroom); selection effectiveness, measured by selector gains over random selection on the same candidate pool; and uncertainty informativeness, measured by how well dispersion in sampled actions predicts failure. Our results show additional samples expose substantial oracle headroom, reaching nearly 50 percent for Astra with a waypoint scaffold on LIBERO, but tested selectors fail to achieve the same headroom consistently. We also diagnose what is behind this action selection failure and discuss future directions for WM research. (Anonymous project page: `https://robot-tts.github.io/`)

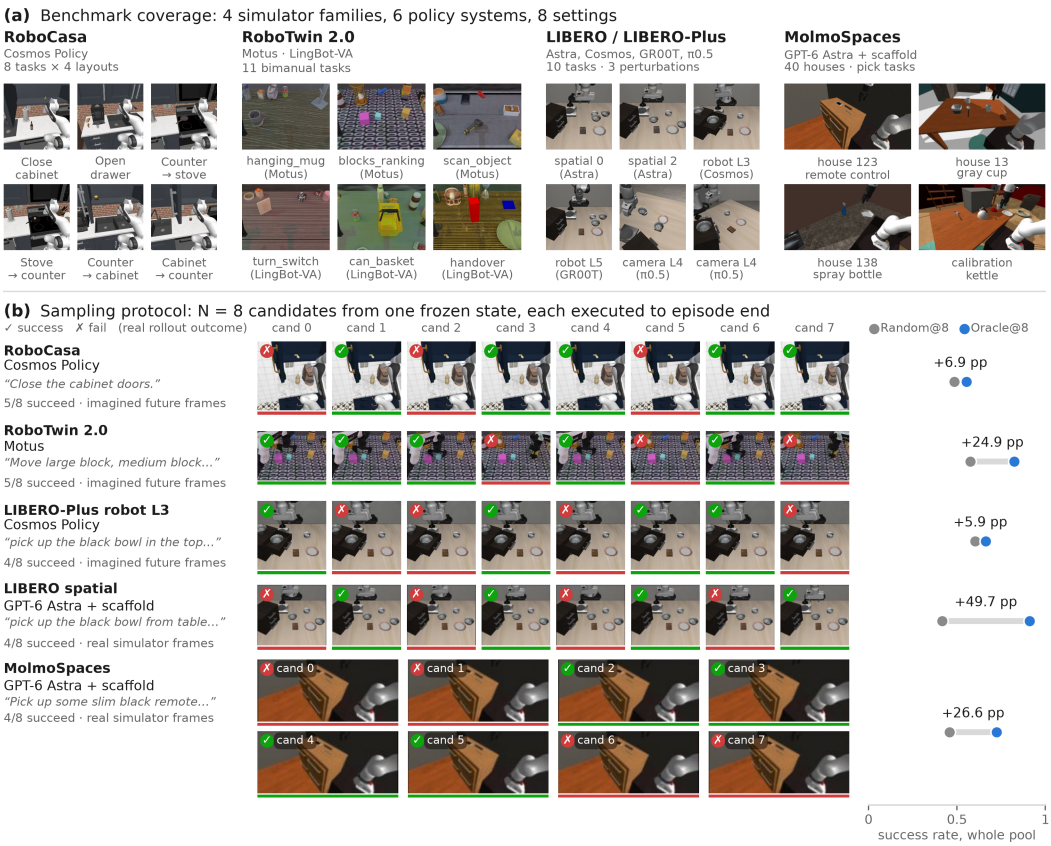
1 INTRODUCTION

Robot foundation policies, particularly vision-language-action (VLA) models, are becoming increasingly capable at taking robot actions based on visual observations and language instructions (Zitkovich et al., 2023; Kim et al., 2025; Black et al., 2025; Physical Intelligence et al., 2025). Alongside these advances, test-time scaling has attracted attention as a way to improve performance by allocating additional inference computation in robot policies (Snell et al., 2025; Kwok et al., 2025; Li et al., 2026b). Specifically, rather than executing a single sampled action, a policy can generate several candidates and select among them (Kwok et al., 2025; Jang et al., 2026; Xie et al., 2026).

In parallel, action-conditioned World Models (WMs) are becoming increasingly capable at predicting the consequences of robot actions and supporting planning in learned visual representations (Zhou et al., 2025; Assran et al., 2025). This naturally leads to using WMs to help select actions generated by a VLA (Jain et al., 2026). Taking one step further, integrated world-action models (WAMs) combine action generation and future visual prediction (Bi et al., 2026; Li et al., 2026a); some additionally models future states and values (Kim et al., 2026). Both the modular VLA+WM pairings and the integrated WAMs offer a route to test-time scaling: sample more actions, imagine their consequences, and select the best action.

Together, these developments motivate a systematic study of how robot foundation policies behave as the sampling budget grows, particularly how world models help identify the right action. More samples may increase the chance that a successful action is available, but realizing that opportunity requires being able to select it (Xie et al., 2026; Zhao et al., 2026). Imagined futures could make the selection easier, yet inaccurate action consequences (Yang et al., 2026) or unreliable judgments of task completion (Khose et al., 2026) could limit their usefulness for selection. This leads to our central question: *How do robot foundation policies scale with additional test-time samples, and how well do world models help identify successful actions?*

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.



Contribution. We introduce a diagnostic benchmark for sample-based test-time scaling, spanning VLAs, WAMs, modular VLA-WM pairings, and GPT-6 Astra operated through a waypoint scaffold. Our contribution is a controlled evaluation that separates the opportunity created by sampling from the actual gain that practical selectors can achieve. We organize this evaluation around three core dimensions:

- **Sampling potential (Section 3).** We vary the sample count and measure oracle headroom: the gain over random selection available to a perfect selector on the same candidate pool. Across eight settings, headroom reaches nearly 50 percentage points for Astra with a waypoint scaffold on LIBERO.
- **Selection effectiveness (Section 4).** We evaluate action consensus, WM-assisted VLM judging, and native value-based selection. These selectors provide *little consistent gain over random selection*. To diagnose why, we substitute simulator-generated future images and states at different horizons. We find even accurate short-horizon futures often remain difficult to rank; whereas longer simulator rollouts improve ranking for Motus and GR00T. However, when trying to replicate the benifit on WMs, the tested recursive Motus rollout does not reproduce this benefit. This suggests future WM research should focus on more reliable prediction at decision-relevant long horizons.
- **Uncertainty informativeness (Section 5).** We test whether dispersion among sampled actions predicts failure. Cosmos Policy shows a positive signal across seeds on spatial tasks and in a separate suite-breadth study, but usefulness varies by policy and execution stage. Therefore, despite not being able to reliably identify the best action, more samples can instead inform failure prediction, although this benefit is not universal.

Overall, additional samples expose substantial oracle headroom, but the tested selectors fail to achieve the headroom consistently. Our diagnostics motivate two priorities for WM research: reliable prediction of longer-horizon consequences, and representations (beyond image) that can be readily

adapted for state and task-value prediction. Both directions should be judged by whether they translate additional test-time computation into better action choices (Section 6).

2 Related Work

Test-time sampling and scaling in robot foundation policies. In language models, self-consistency aggregates sampled answers (Wang et al., 2023), while verifier-guided search and adaptive compute allocation provide other routes to test-time scaling (Snell et al., 2025). Recent methods improve VLA inference by generating multiple action candidates and selecting among them. RoboMonkey combines test-time sampling with a learned VLM-based verifier (Kwok et al., 2025), whereas MG-Select obtains a verifier-free selection signal from masked-condition action distributions (Jang et al., 2026). VLA-ATTC uses uncertainty to allocate additional inference computation and a relative action critic to compare candidates (Li et al., 2026b). SVA measures oracle-style sampling headroom and distills simulator tree-search returns into an action evaluator for a frozen VLA (Xie et al., 2026). These are primarily method contributions, with analyses of their scaling behavior, rather than cross-policy benchmark suites. Complementary failure-monitoring methods include Sentinel, which combines temporal action-consistency checks with VLM assessment, and FAIL-Detect, which calibrates uncertainty signals for runtime detection (Agia et al., 2025; Xu et al., 2025). Our benchmark separates successful-candidate generation, candidate selection, and failure prediction across policy families and decision signals.

Test-time sampling and scaling with world models and WAMs. World models enable candidates to be evaluated through their predicted consequences, using either a modular policy–world-model system or an integrated world-action model (WAM). DINO-WM and V-JEPA 2-AC plan through action-conditioned latent predictions (Zhou et al., 2025; Assran et al., 2025); Motus and LingBot-VA integrate visual prediction and action generation (Bi et al., 2026; Li et al., 2026a). WEAVER combines policy proposals with latent-future prediction and reward-based scoring (Jain et al., 2026), while Cosmos Policy jointly models actions, future observations, and values and uses them for best-of-N planning (Kim et al., 2026). GeoBoN ranks sampled WAM rollouts through geometric consistency and analyzes selection errors as the candidate budget increases (Zhao et al., 2026). In a closely related diagnostic study, Ruan et al. (2026) examine the relationship between action–state consistency and task success, and propose selection by consensus among imagined futures. These are method contributions and diagnostic studies, rather than dedicated cross-family benchmark suites; they already investigate selection bottlenecks and the value of future-based evidence. Beyond selection, Foresight uses action-conditioned world-model latents for failure detection (Zhang et al., 2026). Our benchmark brings these two uses of predictions into a common evaluation: whether imagined outcomes help identify a successful candidate, and whether dispersion across the outcomes of different sampled actions helps anticipate failure.

Benchmarks for robot policies and world models. Existing benchmark suites evaluate complementary properties of policies and predictive models. LIBERO studies knowledge transfer in lifelong manipulation (Liu et al., 2023); VLABench evaluates language-conditioned manipulation and long-horizon reasoning (Zhang et al., 2025); and LIBERO-Plus probes VLA robustness under controlled perturbations (Fei et al., 2025). RoboCasa and RoboTwin 2.0 provide diverse household and bimanual manipulation tasks (Nasiriany et al., 2024; Chen et al., 2025), while SIMPLER studies how well simulated evaluations reflect real-world policy behavior (Li et al., 2025b). WorldModel-Bench evaluates instruction following and physical plausibility in generated videos (Li et al., 2025a), and WorldSimBench combines perceptual and action-level evaluation (Qin et al., 2025). For world models, WorldArena evaluates both perceptual quality and functional utility as data generators, policy evaluators, and planners (Shang et al., 2026). RoboWM-Bench tests whether generated manipulation videos translate into executable robot behavior (Jiang et al., 2026), while MiraBench diagnoses physical consistency, action fidelity, and optimism bias (Yang et al., 2026). WFM-Eval further examines task-completion judging and structured object-level failures (Khose et al., 2026). Thus, prior benchmarks already extend beyond visual realism to embodied usefulness and predictive reliability. Our emphasis is complementary: sample-based test-time scaling across VLA policies, integrated WAMs, and modular systems. We measure how candidate budgets affect oracle success and practical selection, and separately test the failure-prediction value of action- and future-based dispersion. This

organization connects sampling potential, selection effectiveness, and uncertainty informativeness without assuming that stronger performance on one implies stronger performance on the others.

3 SAMPLING POTENTIAL

Key question. *Do additional test-time samples lead to improved performance, assuming a perfect oracle selector is available?*

We first measure the opportunity created by sampling, before evaluating practical selection and failure prediction in Sections 4 and 5. Our studies span six policy systems in eight settings across four simulator families: LIBERO (including LIBERO-Plus), RoboCasa, RoboTwin, and MolmoSpaces.

Oracle Headroom Protocol. At a decision state s, we sample N candidate action chunks and execute each from the same reset or replayed state, followed by a closed-loop policy continuation, to obtain a binary task outcome $r_i(s)$. For M evaluated states $\mathcal{S}$, oracle and uniform-random selection success are

$$O_N = \frac{1}{M} \sum_{s \in \mathcal{S}} \max_{i \leq N} r_i(s), \qquad R_N = \frac{1}{M} \sum_{s \in \mathcal{S}} \frac{1}{N} \sum_{i=1}^{N} r_i(s). \tag{1}$$

Oracle headroom is defined as $H_N = O_N - R_N$. Full protocols are in Appendix B.

Table 1: **Oracle headroom across candidate budgets.** The $N = 8$ column includes 95% intervals. Complete intervals at every budget and Astra's all-subset estimates are in Appendix B.

Policy / simulator	Ep.	H_2	H_4	H_8 [95% CI]
Astra + scaffold / LIBERO	80	20.6	40.3	49.7 [42.2, 57.3]
Motus / RoboTwin	120	9.2	21.7	24.9 [19.3, 33.4]
LingBot-VA / RoboTwin	120	7.9	12.1	16.7 [13.4, 20.2]
Cosmos Policy / RoboCasa	384	3.3	5.9	6.9 [3.7, 10.0]
Cosmos Policy / LIBERO-Plus robot L3	80	3.8	5.0	5.9 [1.7, 10.6]
Astra + scaffold / MolmoSpaces	40	8.8	20.6	26.6 [16.9, 36.6]
GR00T-N1.7 / LIBERO-Plus robot L5	80	8.8	12.2	16.1 [8.3, 24.0]
$\pi_{0.5}$ / LIBERO-Plus camera L4	80	5.6	8.1	11.4 [1.9, 26.9]

More samples create opportunity, with setting-dependent gains. Figure 1 shows different scaling profiles. Motus gains most of its headroom by $N = 4$, whereas Astra and LingBot-VA continue gaining over larger N; Cosmos Policy has smaller absolute gains. Overall, these results show consistent oracle headroom across model families and settings.

At what time step is sampling most useful? The five scans of oracle headroom over different time steps show different temporal patterns (Table 2). Motus has a clear late decline, whereas Cosmos Policy / RoboCasa retains positive headroom without a statistically detected endpoint decrease. Astra detects no change in its limited early window. LingBot-VA and Cosmos Policy / LIBERO-Plus also show lower late headroom. Together, these results do not establish a common rule for allocating more samples.

Key takeaway. Additional samples expose substantial oracle headroom, but its magnitude and temporal pattern depend on the policy system, environment, and execution protocol.

4 SELECTION EFFECTIVENESS

Key question. *Can practical selectors exploit the sampling opportunity, and how well do WMs help in identifying successful actions?*

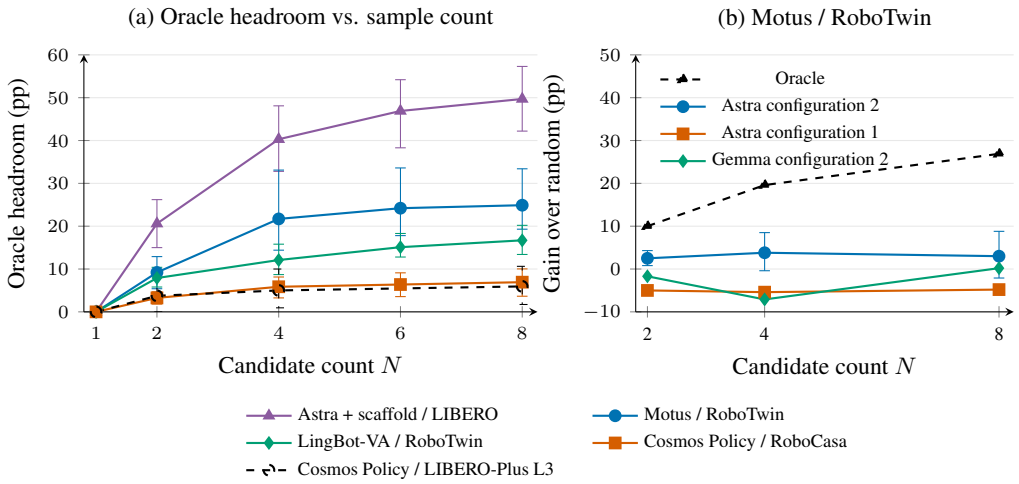


Figure 1: **Sampling opportunity and practical selection.** Left: oracle headroom for the five primary settings, with 95% intervals. Right: **Motus / RoboTwin**; its oracle and judge curves remain on that same pool. Whiskers show Astra configuration 2 intervals. Full numerical results: Appendix B and Table 10.

Table 2: **Temporal persistence of oracle headroom.** Oracle headroom ($N = 4$, pp) evaluated at progressively later checkpoints. ΔH compares the final checkpoint against the same cohort's $j = 0$ reference. Full confidence intervals and sensitivity analyses are in Appendix B.2.

Policy / simulator	**Cohort**	**Oracle headroom trajectory** $j{:}H_j$	ΔH_{final} **vs.** $j{=}0$
Motus / RoboTwin	120 / 67	0:25.4 → 1:21.6 → 2:19.4 → 4:16.4 → 8:4.1	−21.3 [−35.3, −10.2]
Cosmos Policy / RoboCasa	288 / 254	0:5.4 → 4:6.7 → 8:3.2 → 12:3.2	−2.26 [−5.68, +0.61]
Astra + scaffold / LIBERO	20 / 20	4:36.2 → 8:37.5 → 12:36.3	−0.6 [−13.7, +12.7]
LingBot-VA / RoboTwin	72 / 32	0:17.2 → 2:18.8 → 4:7.0	−10.2 [−22.4, 0.0]
Cosmos Policy / LIBERO-Plus robot L3	80 / 30	0:3.3 → 4:0.8 → 8:0.0 → 12:0.0	−3.3 [−10.7, 0.0]

4.1 Selection baselines

Section 3 establishes the gain over random available from sampled actions. We now test three ways to select among sampled actions.

- **Action consensus:** This is the continuous-action analogue of majority voting used in LLMs, where we select a candidate near the sample mean, a medoid, or a dense region in action space. This method uses only the sampled actions, without a WM.
- **Modular WM + VLM selection:** an action-conditioned WM predicts each candidate's future, which a VLM ranks for likely success. We test Cosmos-Policy-WM and, separately, corrected WAP predictions; the VLM part uses Gemma-4-31B and GPT-6 Astra judges.
- **Integrated WAM selection:** We use VLMs to judge the image generated by WAMs (that jointly generate the image and action), as in Motus. We also use the predicted value estimate to select action, as in Cosmos Policy.

Metric. Our main metric is selection gain, $G_N = S_N - R_N$: the difference between the selector's success rate and random-selection success on the same pool of actions. For diagnosis, we also report within-pool *pairwise accuracy*: ranking successful above failed candidates, with ties counting as 0.5 (chance). Full protocols are in Appendix C.2.

Table 3: **Selection effectiveness across practical selectors and diagnostic controls.** Panel A gives selection gain over random in pp; Panels B–C give pairwise accuracy (chance 0.5). Panel B reports **Gemma / Astra** image judges separately; Panel C uses learned classifiers on true simulator states. t is environment steps after the branch.

A. Practical selectors and oracle reference: selection gain (pp)

Policy / simulator	Selector	N	Gain	95% CI
Motus / RoboTwin	Consensus	8	+0.2	—
LingBot / RoboTwin	Consensus	8	+0.5	$[-6.8, 7.6]$
Cosmos Policy / RoboCasa	Oracle	8	$+6.93$	$[3.65, 10.03]$
Cosmos Policy / RoboCasa	Native value	8	-1.40	$[-3.52, 0.46]$
Cosmos Policy / RoboCasa	Gemma VLM	8	-0.88	$[-2.21, 0.52]$

B. Image judges: pairwise accuracy (Gemma / Astra)

Policy / simulator	**Imagined**	**Simulator: short**	**Simulator: long**
Motus / RoboTwin	.534 / .521	.523 / .606 ($t = 16$)	.605 / .746 ($t = 128$)
GR00T / LIBERO-Plus	.521 / .542	.458 / .484 ($t = 16$)	.276 / .797 ($t = 256$)
$\pi_{0.5}$ / LIBERO-Plus	.500 / .604	.448 / .470 ($t = 10$)	.485 / .470 ($t = 40$)
Astra policy / LIBERO	.505 / .359	.505 / .578 ($t = 10$)	.520 / .460* ($t = 40$)
LingBot / RoboTwin	— / —	.521 / .405 ($t = 16$)	.513 / .581 ($t = 64$)

C. Privileged-state linear classifiers: pairwise accuracy

Policy / simulator	State accuracy (lookahead t)
Motus / RoboTwin	.66 ($t = 16$); .80 ($t = 128$)
GR00T / LIBERO-Plus	.22 ($t = 8$); .90 ($t = 256$)
$\pi_{0.5}$ / LIBERO-Plus	.49–.58 across $t = 5$–80
Astra policy / LIBERO	.42 ($t = 16$); .68 ($t = 40$)

4.2 Selection results

Action consensus provides little reliable selection benefit. Consensus gains are small in the reported Motus pool and LingBot results (Table 3A). In LingBot, 19 of 22 mixed-outcome pools have a successful majority: at least five of the eight candidates succeed (62.5% or more). Yet choosing the action closest to the mean gains only 0.5 pp and has pairwise accuracy 0.487. Thus, a successful majority need not correspond to the center of action space. Results on variations of the consensus rules and action representation sweeps are reported in Appendix C.1.

Short-horizon imagined futures do not reliably select better than random. We next ask whether imagined futures from WMs provide information that action consensus misses. The image-based comparison covers five policies across RoboTwin, LIBERO, and LIBERO-Plus (Table 3B). Motus supplies joint action–video predictions, while GR00T and the Astra policy use Cosmos-Policy-WM. Neither Gemma-4-31B nor GPT-6 Astra shows a reliable positive ranking signal from the reported Motus, GR00T, or $\pi_{0.5}$ imagined futures; Astra judging its own waypoint candidates instead has an inverted signal (0.359). Lastly, jointly generating actions and futures, as in Motus, does not by itself help selection better than random.

Native value-based selection shows no established gain for Cosmos Policy. Instead of asking a VLM to judge a future image, Cosmos Policy (Kim et al., 2026) can rank candidates using its own value estimate. On the same 384-episode RoboCasa pool, neither native value nor Gemma-4-31B judging establishes a gain over random selection, despite positive oracle headroom (Table 3A). The VLM frequently gives all candidates tied scores, limiting its ability to rank them; details are in Appendix C. We note that this setting differs from the positive planning results reported by Kim et al. (2026), which use a rollout-refined world-model/value checkpoint for best-of-N planning on challenging real-world ALOHA tasks.

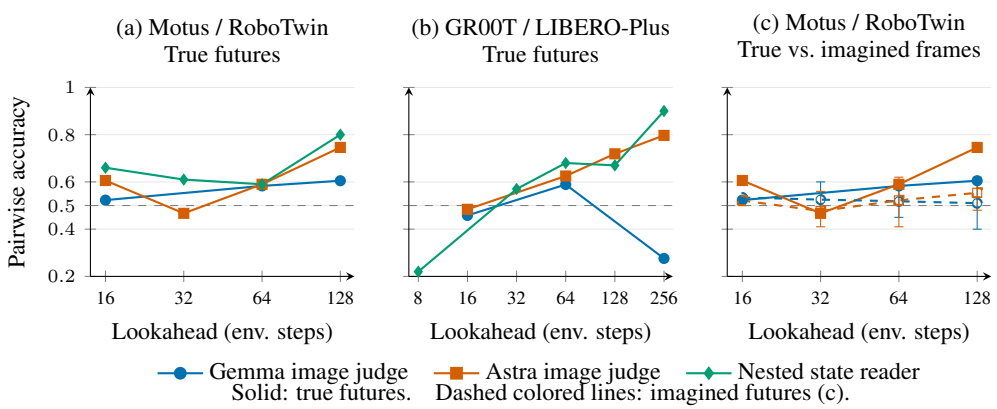


Figure 2: **Longer true futures can help selection; longer imagined rollouts need not.** Left and center: selection based on simulator-image judges and state readers. Right: Motus image judging with real images versus recursively imagined futures. The horizontal axis is lookahead from the same branch. Dashed gray lines mark chance. Full numerical details: Tables 13, 14, and 16.

4.3 Failure diagnostics

Given these mostly ineffective results for action selection, we ask whether selection is limited by inaccurate WM predictions, by the evaluator (VLM judge), or by how far into the future we look. We investigate these possibilities with two privileged controls:

- **VLM on simulator images:** We replace the imagined image with a simulator-rendered future while retaining the VLM judge and prompt.
- **Classifier on simulator states:** We train a classifier model for each task, scoring candidates with a weighted sum of their true future-state features. Training compares successful and unsuccessful candidates from the same decision state and learns to score the successful candidate higher. This bypasses image interpretation. The protocols are detailed in Appendix C.2.

VLM selection from simulator-generated short-horizon images. Are these failures caused by inaccurate WM predictions or by the VLM judge? To test this, we replace the WM-predicted future with a simulator-generated image while keeping the judge unchanged. This does not consistently improve selection (Table 3B): paired real-minus-imagined accuracy differences are $+0.09$ $[-0.08, 0.33]$ for Motus/Astra and -0.06 $[-0.19, 0.08]$ for GR00T/Astra. Motus's real-image accuracy of 0.606 is above chance, but both judges remain near or below chance for GR00T and $\pi_{0.5}$. Thus, even when the short-horizon image is correct, the VLM often cannot tell which action will eventually succeed.

Classifier-based selection from true future states. Perhaps another difficulty is interpreting images. To bypass that step, we train a simple classifier directly on simulator future states (Table 3C). At short horizons, Motus reaches 0.66 accuracy at 16 steps, whereas GR00T's 0.22 at eight steps reverses the desired ranking. For $\pi_{0.5}$, accuracy remains between 0.49 and 0.58 over the reported 5–80-step range. Thus, access to true state does not guarantee useful selection at short horizons.

Longer true rollouts make successful candidates easier to identify. If short horizons are insufficient, can looking farther ahead help? For Motus and GR00T, both Astra's image judgments and the learned state classifiers are more accurate at the longest tested horizons than at the short ones (Figure 2). The better image rankings also translate into positive selection-gain estimates for both policies, although only GR00T's confidence interval excludes zero (Table 15). This benefit is not universal: $\pi_{0.5}$ and LingBot do not show the same clear improvement, and Gemma's ranking worsens for GR00T. These controls suggest that an action's consequences may only become clear farther into the future. They rely on simulator access and sometimes near-end-of-episode observations, so they diagnose what information helps rather than provide a deployable selector.

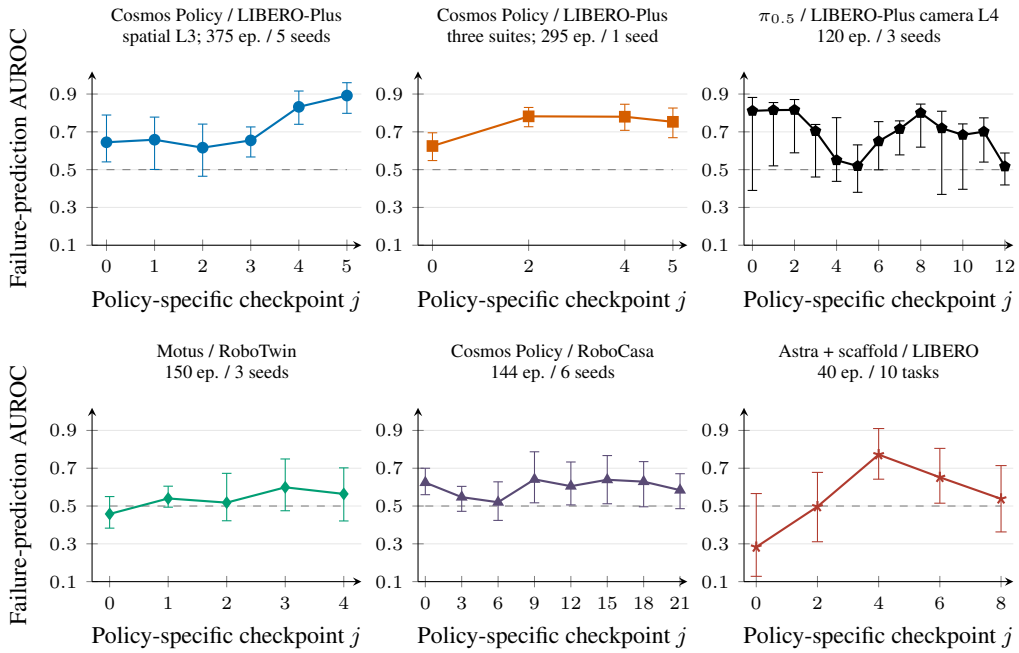


Figure 3: Action dispersion can be informative, but its usefulness is setting- and checkpoint-specific. Bars show reported 95% intervals; dashed lines mark chance. Full values and counts: Table 17.

Longer horizon rollouts in WM do not reproduce this benefit. Knowing that longer true rollouts can improve action selection, we next ask whether WMs can provide the same benefit. We test this on Motus. To roll out recursively, we feed back its predicted image and use the last commanded joint target as approximate proprioception. After eight chunks (128 steps), Astra achieves only 0.554 accuracy, versus 0.746 on real images at the same horizon; the paired imagined-minus-real difference is -0.19 $[-0.30, -0.01]$. This shows the fidelity of long horizon rollouts in our tested WM may not be sufficient for it to be useful for action selection.

Key takeaway. Tested action-consensus and short-horizon WM-imagined futures provide limited selection gains. Longer simulator rollouts improve ranking for Motus and GR00T. However, our recursive Motus rollout does not recover the same benefit.

5 Uncertainty Informativeness

Key question. *When does disagreement among sampled actions predict failure?*

Predicting failure from action dispersion. We measure dispersion among action sequences sampled from the same observation: the mean coordinate-wise standard deviation across samples, averaged over the action sequence. We test whether higher dispersion predicts eventual failure, using AUROC (0.5: chance; 1.0: perfect). Figure 3 evaluates this signal across six policy–simulator settings and multiple execution checkpoints (see Appendix D for more details.)

Failure-prediction informativeness is setting- and checkpoint-specific. Cosmos Policy / LIBERO-Plus provides the clearest positive evidence: action dispersion becomes strongly predictive at later checkpoints in the spatial-task study, and a separate three-suite breadth study shows similarly positive signals across broader task families. Other settings, including $\pi_{0.5}$ / LIBERO-Plus, Motus / RoboTwin, Cosmos Policy / RoboCasa, and Astra + scaffold / LIBERO, show weak informativeness at some checkpoints while remaining noisy at most others. We note that the Astra result is based on a relatively small-sample experiment and is included mainly as supportive evidence;

nevertheless, it shows a strong failure-prediction signal at $j = 4$ (48–50 environment steps), with AUROC 0.772 $[0.642, 0.910]$ and within-task $p = .008$. Overall, the methodology helps identify when action dispersion is useful for failure prediction in a given policy–environment setting.

Do value-based and imagined-future signals help? For Cosmos Policy, value dispersion is also informative for failure prediction at later checkpoints. In the spatial-task study, value-dispersion AUROC reaches 0.831 at $j = 4$ and 0.819 at $j = 5$, with similarly positive signals in the three-suite breadth study. However, value and action dispersion are strongly correlated ($\rho = .757$ at $j = 4$ and $.756$ at $j = 5$), and residual analysis favors action dispersion: at $j = 5$, action dispersion residualized on value remains predictive (AUROC 0.754 $[0.640, 0.847]$), whereas value dispersion residualized on action does not (0.575 $[0.473, 0.683]$). Thus, value dispersion provides a useful secondary failure signal, but we find no established incremental benefit beyond action dispersion. Imagined-future dispersion shows a similar mixed picture: Motus does not retain a clear signal after controlling for action dispersion, while LingBot-VA shows an exploratory positive residual signal. RoboCasa whole-episode value-mean prediction does not reproduce on held-out seeds. Full comparisons are in Appendix D.4.

Key takeaway. Action dispersion can predict failure, but its informativeness is strongly policy- and checkpoint-dependent and stage dependent. Value and imagined-future signals can also contain failure information, but provide no consistent additional benefit beyond action dispersion from our tested cases.

6 Discussion

Our results suggest the following future directions for world models intended to improve robot decision-making through test-time scaling.

Reliable prediction at decision-relevant horizons. True short-horizon futures often remain difficult for the tested selectors to rank, whereas longer simulator rollouts improve Astra's judgments and state-classifier rankings for Motus and GR00T (Section 4.3). Yet extending Motus's imagined rollouts does not recover this benefit. Future WMs should therefore have the capability to predict action-dependent consequences over horizons long enough to distinguish candidate outcomes. Multi-horizon prediction and consistent visual and proprioceptive modeling are directions worth exploring.

Representations that are easy to adapt for value prediction. Our results show that simulated future states can be more effective at helping with action selection compared to images in some test cases. Therefore, beyond predicting future images, WMs should learn representations that are state and task relevant and can be efficiently fine-tuned to estimate the task value of candidate actions. A lightweight, task-conditioned value head could learn from successful and failed continuations using action-conditioned WM features, without repeatedly asking a general-purpose VLM to interpret generated frames.

Decision-relevant diversity and self-evaluation. Test-time scaling is useful only when additional samples produce meaningfully different, task-relevant actions and futures. Preserving such diversity may also improve robustness across diverse situations, but diversity alone is insufficient: the model must identify which candidates are promising or failure-prone. The coexistence of substantial oracle headroom with weak practical selection in our results suggests that future policies and WAMs should pair diverse sampling with stronger native value heads or external verifiers trained explicitly for candidate discrimination.

AI USE STATEMENT

Generative AI tools were used to build the code base of the benchmark, convert the raw data into tables and figures in the paper. The authors reviewed all AI-assisted contents and take responsibility for the final content of this work, including text, claims, and artifacts produced with the aid of generative AI.

REPRODUCIBILITY STATEMENT

The evaluation protocols, cohort definitions, statistical procedures, and key robustness checks are described in the main text and Appendices B, C, and D. The appendices provide complete numerical counterparts to the main-text figures and tables.

REFERENCES

Christopher Agia, Rohan Sinha, Jingyun Yang, Ziang Cao, Rika Antonova, Marco Pavone, and Jeannette Bohg. Unpacking failure modes of generative policies: Runtime monitoring of consistency and progress. In *Proceedings of The 8th Conference on Robot Learning*, volume 270 of *Proceedings of Machine Learning Research*, pp. 689–723. PMLR, 2025. URL https://proceedings.mlr.press/v270/agia25a.html.

Mahmoud Assran, Adrien Bardes, David Fan, Quentin Garrido, Russell Howes, Mojtaba Komeili, Matthew Muckley, Ammar Rizvi, Claire Roberts, Koustuv Sinha, Artem Zholus, Sergio Arnaud, Abha Gejji, Ada Martin, Francois Robert Hogan, Daniel Dugas, Piotr Bojanowski, Vasil Khalidov, Patrick Labatut, Francisco Massa, Marc Szafraniec, Kapil Krishnakumar, Yong Li, Xiaodong Ma, Sarath Chandar, Franziska Meier, Yann LeCun, Michael Rabbat, and Nicolas Ballas. V-JEPA 2: Self-supervised video models enable understanding, prediction and planning. *arXiv preprint arXiv:2506.09985*, 2025. URL https://arxiv.org/abs/2506.09985.

Hongzhe Bi, Hengkai Tan, Shenghao Xie, Zeyuan Wang, Shuhe Huang, Haitian Liu, Ruowen Zhao, Yao Feng, Chendong Xiang, Yinze Rong, Hongyan Zhao, Hanyu Liu, Zhizhong Su, Lei Ma, Hang Su, and Jun Zhu. Motus: A unified latent action world model. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 35101–35113, 2026. URL https://openaccess.thecvf.com/content/CVPR2026/html/Bi_Motus_A_Unified_Latent_Action_World_Model_CVPR_2026_paper.html.

Kevin Black, Noah Brown, Danny Driess, Adnan Esmail, Michael Robert Equi, Chelsea Finn, Niccolo Fusai, Lachy Groom, Karol Hausman, Brian Ichter, Szymon Jakubczak, Tim Jones, Liyiming Ke, Sergey Levine, Adrian Li-Bell, Mohith Mothukuri, Suraj Nair, Karl Pertsch, Lucy Xiaoyang Shi, Laura Smith, James Tanner, Quan Vuong, Anna Walling, Haohuan Wang, and Ury Zhilinsky. π_0: A vision-language-action flow model for general robot control. In *Proceedings of Robotics: Science and Systems*, 2025. doi: 10.15607/RSS.2025.XXI.010. URL https://www.roboticsproceedings.org/rss21/p010.html.

Tianxing Chen, Zanxin Chen, Baijun Chen, Zijian Cai, Yibin Liu, Zixuan Li, Qiwei Liang, Xianliang Lin, Yiheng Ge, Zhenyu Gu, Weiliang Deng, Yubin Guo, Tian Nian, Xuanbing Xie, Qiangyu Chen, Kailun Su, Tianling Xu, Guodong Liu, Mengkang Hu, Huan-ang Gao, Kaixuan Wang, Zhixuan Liang, Yusen Qin, Xiaokang Yang, Ping Luo, and Yao Mu. RoboTwin 2.0: A scalable data generator and benchmark with strong domain randomization for robust bimanual robotic manipulation. *arXiv preprint arXiv:2506.18088*, 2025. URL https://arxiv.org/abs/2506.18088.

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: In-depth robustness analysis of vision-language-action models. *arXiv preprint arXiv:2510.13626*, 2025. URL https://arxiv.org/abs/2510.13626.

Arnav Kumar Jain, Yilin Wu, Jesse Farebrother, Gokul Swamy, and Andrea Bajcsy. WEAVER, better, faster, longer: An effective world model for robotic manipulation. *arXiv preprint arXiv:2606.13672*, 2026. URL https://arxiv.org/abs/2606.13672.

Suhyeok Jang, Dongyoung Kim, Changyeon Kim, Youngsuk Kim, and Jinwoo Shin. Verifier-free test-time sampling for vision-language-action models. In *The Fourteenth International Conference on Learning Representations*, 2026. URL https://openreview.net/forum?id=UD4Rw8MOEK.

Feng Jiang, Yang Chen, Kyle Xu, Yuchen Liu, Haifeng Wang, Zhenhao Shen, Jasper Lu, Shengze Huang, Yuanfei Wang, Chen Xie, and Ruihai Wu. RoboWM-Bench: A benchmark for evaluating world models in robotic manipulation. *arXiv preprint arXiv:2604.19092*, 2026. URL https://arxiv.org/abs/2604.19092.

Sahil Khose, Mengqi Zhang, Prithvijit Chattopadhyay, and Judy Hoffman. WFM-Eval: An evaluation framework for video world models in robotic manipulation. In *CVPR 2026 Workshops on Video World Models and Foundation Models Meet Embodied Agents*, 2026. URL https://sahilkhose.github.io/wfm-eval-cvpr26/.

Moo Jin Kim, Karl Pertsch, Siddharth Karamcheti, Ted Xiao, Ashwin Balakrishna, Suraj Nair, Rafael Rafailov, Ethan P. Foster, Pannag R. 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 *Proceedings of The 8th Conference on Robot Learning*, volume 270 of *Proceedings of Machine Learning Research*, pp. 2679–2713. PMLR, 2025. URL https://proceedings.mlr.press/v270/kim25c.html.

Moo Jin Kim, Yihuai Gao, Tsung-Yi Lin, Yen-Chen Lin, Yunhao Ge, Grace Lam, Percy Liang, Shuran Song, Ming-Yu Liu, Chelsea Finn, and Jinwei Gu. Cosmos policy: Fine-tuning video models for visuomotor control and planning. In *International Conference on Learning Representations*, volume 2026, pp. 71531–71552, 2026. URL https://proceedings.iclr.cc/paper_files/paper/2026/file/748becc400a57c0e31cfe6a2e7951467-Paper-Conference.pdf.

Jacky Kwok, Christopher Agia, Rohan Sinha, Matt Foutter, Shulu Li, Ion Stoica, Azalia Mirhoseini, and Marco Pavone. RoboMonkey: Scaling test-time sampling and verification for vision-language-action models. *arXiv preprint arXiv:2506.17811*, 2025. URL https://arxiv.org/abs/2506.17811.

Dacheng Li, Yunhao Fang, Yukang Chen, Shuo Yang, Shiyi Cao, Justin Wong, Michael Luo, Xiaolong Wang, Hongxu Yin, Joseph E. Gonzalez, Ion Stoica, Song Han, and Yao Lu. WorldModelBench: Judging video generation models as world models. *arXiv preprint arXiv:2502.20694*, 2025a. URL https://arxiv.org/abs/2502.20694.

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. Causal world modeling for robot control. *arXiv preprint arXiv:2601.21998*, 2026a. URL https://arxiv.org/abs/2601.21998.

Wenhao Li, Xiu Su, Yichao Cao, Hongyan Xu, Xiaobo Xia, Shan You, Yi Chen, and Chang Xu. VLA-ATTC: Adaptive test-time compute for VLA models with relative action critic model. *arXiv preprint arXiv:2605.01194*, 2026b. URL https://arxiv.org/abs/2605.01194.

Xuanlin Li, Kyle Hsu, Jiayuan Gu, Oier Mees, Karl Pertsch, Homer Rich Walke, Chuyuan Fu, Ishikaa Lunawat, Isabel Sieh, Sean Kirmani, Sergey Levine, Jiajun Wu, Chelsea Finn, Hao Su, Quan Vuong, and Ted Xiao. Evaluating real-world robot manipulation policies in simulation. In *Proceedings of The 8th Conference on Robot Learning*, volume 270 of *Proceedings of Machine Learning Research*, pp. 3705–3728. PMLR, 2025b. URL https://proceedings.mlr.press/v270/li25c.html.

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*, volume 36, 2023. URL https://proceedings.neurips.cc/paper_files/paper/2023/hash/8c3c666820ea055a77726d66fc7d447f-Abstract-Datasets_and_Benchmarks.html.

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 *Proceedings of Robotics: Science and Systems*, 2024. URL https://arxiv.org/abs/2406.02523.

Physical Intelligence, Kevin Black, Noah Brown, James Darpinian, Karan Dhabalia, Danny Driess, Adnan Esmail, Michael Equi, Chelsea Finn, Niccolo Fusai, Manuel Y. Galliker, Dibya Ghosh, Lachy Groom, Karol Hausman, Brian Ichter, Szymon Jakubczak, Tim Jones, Liyiming Ke, Devin LeBlanc, Sergey Levine, Adrian Li-Bell, Mohith Mothukuri, Suraj Nair, Karl Pertsch, Allen Z. Ren, Lucy Xiaoyang Shi, Laura Smith, Jost Tobias Springenberg, Kyle Stachowicz, James Tanner, Quan Vuong, Homer Walke, Anna Walling, Haohuan Wang, Lili Yu, and Ury Zhilinsky. $\pi_{0.5}$: A vision-language-action model with open-world generalization. *arXiv preprint arXiv:2504.16054*, 2025. URL https://arxiv.org/abs/2504.16054.

Yiran Qin, Zhelun Shi, Jiwen Yu, Xijun Wang, Enshen Zhou, Lijun Li, Zhenfei Yin, Xihui Liu, Lu Sheng, Jing Shao, Lei Bai, and Ruimao Zhang. WorldSimBench: Towards video generation models as world simulators. In *Proceedings of the 42nd International Conference on Machine Learning*, volume 267 of *Proceedings of Machine Learning Research*, pp. 50338–50362. PMLR, 2025. URL https://proceedings.mlr.press/v267/qin25f.html.

Bo-Kai Ruan, Teng-Fang Hsiao, Ling Lo, and Hong-Han Shuai. Is the future compatible? diagnosing dynamic consistency in world action models. *arXiv preprint arXiv:2605.07514*, 2026. URL https://arxiv.org/abs/2605.07514.

Yu Shang, Zhuohang Li, Yiding Ma, Weikang Su, Xin Jin, Ziyou Wang, Lei Jin, Xin Zhang, Yinzhou Tang, Haisheng Su, Chen Gao, Wei Wu, Xihui Liu, Dhruv Shah, Zhaoxiang Zhang, Zhibo Chen, Jun Zhu, Yonghong Tian, Tat-Seng Chua, Wenwu Zhu, and Yong Li. WorldArena: A unified benchmark for evaluating perception and functional utility of embodied world models. *arXiv preprint arXiv:2602.08971*, 2026. URL https://arxiv.org/abs/2602.08971.

Charlie Snell, Jaehoon Lee, Kelvin Xu, and Aviral Kumar. Scaling LLM test-time compute optimally can be more effective than scaling parameters for reasoning. In *International Conference on Learning Representations*, 2025. URL https://openreview.net/forum?id=4FWAwZtd2n.

Xuezhi Wang, Jason Wei, Dale Schuurmans, Quoc V. Le, Ed H. Chi, Sharan Narang, Aakanksha Chowdhery, and Denny Zhou. Self-consistency improves chain of thought reasoning in language models. In *International Conference on Learning Representations*, 2023. URL https://openreview.net/forum?id=1PL1NIMMrw.

Xinyi Xie, Zican Hu, Zhanyu Liu, Yicheng Dong, Wenhao Wu, Zhenhong Sun, Haoran Li, Chunlin Chen, Zhi Wang, and Pichao Wang. Look before you leap: Distilling tree search into action evaluation for frozen VLA models. *arXiv preprint arXiv:2607.03751*, 2026. URL https://arxiv.org/abs/2607.03751.

Chen Xu, Tony Khuong Nguyen, Emma Dixon, Christopher Rodriguez, Patrick Miller, Robert Lee, Paarth Shah, Rares Andrei Ambrus, Haruki Nishimura, and Masha Itkina. Can we detect failures without failure data? uncertainty-aware runtime failure detection for imitation learning policies. In *Proceedings of Robotics: Science and Systems*, 2025. doi: 10.15607/RSS.2025.XXI.073. URL https://www.roboticsproceedings.org/rss21/p073.html.

Tianzhuo Yang, Zihan Shen, Zirui Mi, Zhaoyi Zhang, Jiayi Zhou, Jiaming Ji, Juntao Dai, Jiawei Chen, Boyuan Chen, and Yaodong Yang. MiraBench: Evaluating action-conditioned reliability in robotic world models. *arXiv preprint arXiv:2605.29360*, 2026. URL https://arxiv.org/abs/2605.29360.

Haoran Zhang, Yifu Lu, Boyang Wang, Xuhui Kang, Yen-Ling Kuo, Zezhou Cheng, Mengdi Wang, and Odest Chadwicke Jenkins. Foresight: Failure detection for long-horizon robotic manipulation with action-conditioned world model latents. *arXiv preprint arXiv:2606.23085*, 2026. URL https://arxiv.org/abs/2606.23085.

Shiduo Zhang, Zhe Xu, Peiju Liu, Xiaopeng Yu, Yuan Li, Qinghui Gao, 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 *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 11142–11152, 2025. URL https://openaccess.thecvf.com/content/ICCV2025/html/Zhang_VLABench_A_Large-Scale_Benchmark_for_Language-Conditioned_Robotics_Manipulation_with_Long-Horizon_ICCV_2025_paper.html.

Zesen Zhao, Minkyoung Cho, Hui Shen, Boyuan Zheng, Kunxiao Gao, Yulong Cao, and Z. Morley Mao. Test-time scaling for world action models via zero-shot geometric evaluation. *arXiv preprint arXiv:2607.17454*, 2026. URL https://arxiv.org/abs/2607.17454.

Gaoyue Zhou, Hengkai Pan, Yann LeCun, and Lerrel Pinto. DINO-WM: World models on pre-trained visual features enable zero-shot planning. In *Proceedings of the 42nd International Conference on Machine Learning*, volume 267 of *Proceedings of Machine Learning Research*, pp. 79115–79135. PMLR, 2025. URL https://proceedings.mlr.press/v267/zhou25t.html.

Brianna Zitkovich, Tianhe Yu, Sichun Xu, Peng Xu, Ted Xiao, Fei Xia, Jialin Wu, Paul Wohlhart, Stefan Welker, Ayzaan Wahid, Quan Vuong, Vincent Vanhoucke, Huong Tran, Radu Soricut, Anikait Singh, Jaspiar Singh, Pierre Sermanet, Pannag R. Sanketi, Grecia Salazar, Michael S. Ryoo, Krista Reymann, Kanishka Rao, Karl Pertsch, Igor Mordatch, Henryk Michalewski, Yao Lu, Sergey Levine, Lisa Lee, Tsang-Wei Edward Lee, Isabel Leal, Yuheng Kuang, Dmitry Kalashnikov, Ryan Julian, Nikhil J. Joshi, Alex Irpan, Brian Ichter, Jasmine Hsu, Alexander Herzog, Karol Hausman, Keerthana Gopalakrishnan, Chuyuan Fu, Pete Florence, Chelsea Finn, Kumar Avinava Dubey, Danny Driess, Tianli Ding, Krzysztof Marcin Choromanski, Xi Chen, Yevgen Chebotar, Justice Carbajal, Noah Brown, Anthony Brohan, Montserrat Gonzalez Arenas, and Kehang Han. 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.

SUPPLEMENTARY MATERIAL

A EXPERIMENT OVERVIEW

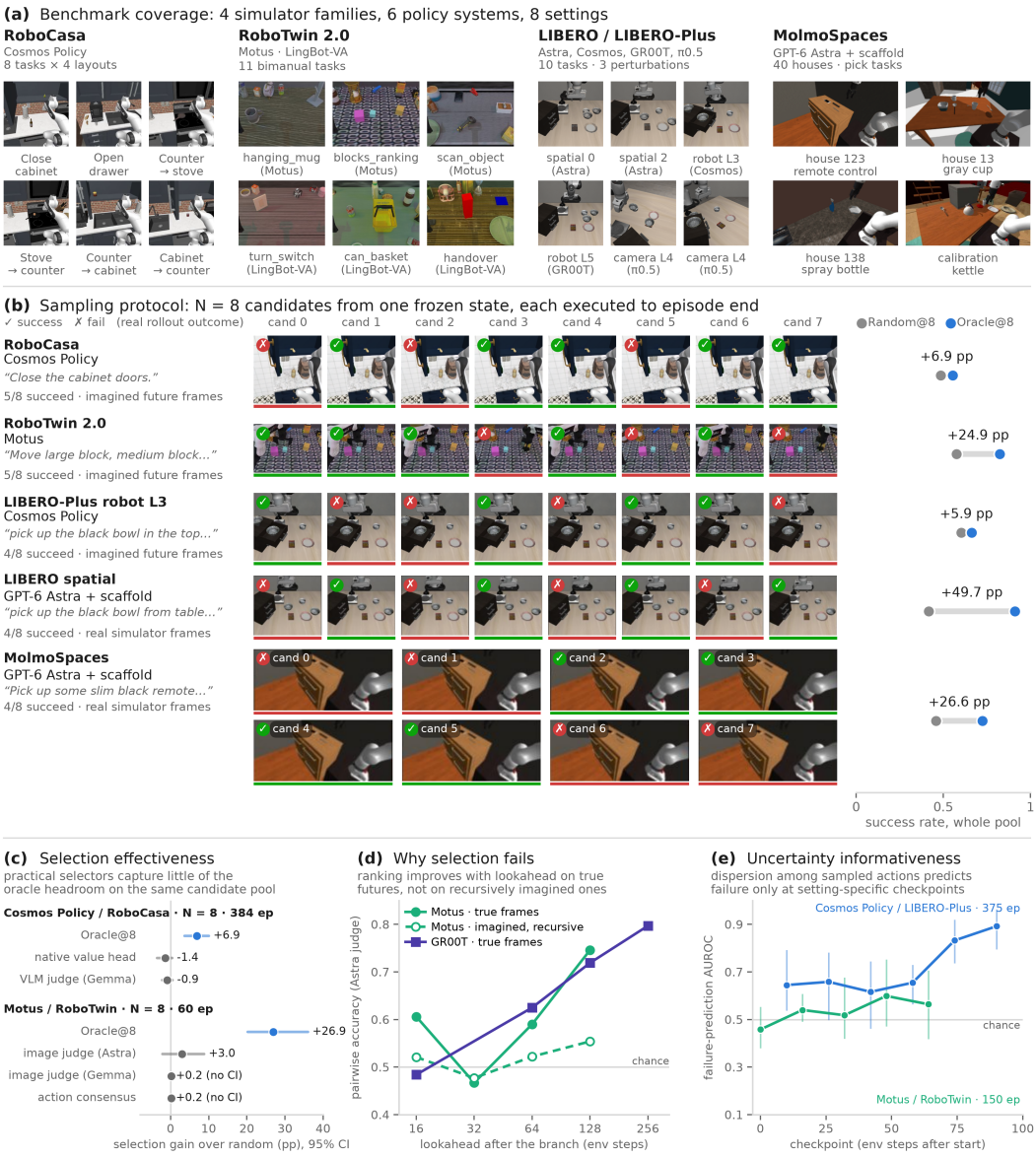


Table 4 summarizes the cohorts used for the main comparisons. Result-specific protocols and statistical details follow in Appendices B–D.

Table 4: **Cohorts and sampling budgets.** Tasks denotes task/base-task clusters, except MolmoSpaces (40 houses in one picking benchmark). Seeds gives the number of evaluation seeds unless marked † (collection labels, not independent seed replications). Episodes/roots are collected counts; arrows identify matched or fixed analysis cohorts, not additional data. LP abbreviates LIBERO-Plus; robot L3/L5 and camera L4 denote perturbation conditions. Complete checkpoint denominators appear in the result tables.

Policy / simulator	Tasks	Seeds	Episodes	N or K	Decision / lookahead
Sampling potential: static candidate pools					
Astra + scaffold / LIBERO	10	2†	80	$N \leq 8$	Initial root
Motus / RoboTwin	5	6	120	$N \leq 8$	Initial root
LingBot-VA / RoboTwin	6	5	120	$N \leq 8$	Initial root
Cosmos Policy / RoboCasa	8	3	384	$N \leq 8$	Initial root
Cosmos Policy / LP robot L3	10	2	80	$N \leq 8$	Initial root
GR00T-N1.7 / LP robot L5	10	2	80	$N \leq 8$	Initial root
$\pi_{0.5}$ / LP camera L4	5	2	80	$N \leq 8$	Initial root
Astra + scaffold / MolmoSpaces	—	1†	40	$N \leq 8$	Initial root
Sampling potential: temporal scans					
Motus / RoboTwin	5	3	$120 \rightarrow 67$	$N = 4$	$j = 0, 1, 2, 4, 8$
Cosmos Policy / RoboCasa	8	3	$288 \rightarrow 254$	$N = 4$	$j = 0, 4, 8, 12$
Astra + scaffold / LIBERO	10	1†	20	$N = 4$	$j = 4, 8, 12$
LingBot-VA / RoboTwin	6	3	$72 \rightarrow 32$	$N = 4$	$j = 0, 2, 4$
Cosmos Policy / LP robot L3	10	2	$80 \rightarrow 30$	$N = 4$	$j = 0, 4, 8, 12$
Selection effectiveness: diagnostic candidate pools					
Motus / RoboTwin	5	3	60	$N = 8$	$t = 16, 32, 64, 128$
GR00T-N1.7 / LP robot L3–L5	10	1†	112	$N = 4$	Images: $t = 16, 64, 128, 256$; states: 8–256
$\pi_{0.5}$ / LP camera L3–L4	8	1†	80	$N = 4$	Images: $t = 10, 40$; states: 5–80
Astra + scaffold / LIBERO	10	2†	40	$N = 4$	Images: $t = 10, 16, 40$; states: $16, 40$
LingBot-VA / RoboTwin	6	2	48	$N = 8$	Images: $t = 16, 32, 64$; states: 64
Cosmos Policy / RoboCasa	8	3	384	$N = 8$	Native value / VLM; same static pool
Cosmos Policy / LP robot L3	10	2	80	$N = 8$	Native value; same static pool
Uncertainty informativeness: passive monitoring					
Cosmos Policy / LP spatial, robot L3	10	5	375	$K = 4$	$j = 0, \ldots, 5$
Cosmos Policy / LP object, goal, long-horizon	30	1	$299 \rightarrow 295$	$K = 4$	$j = 0, 2, 4, 5$
Cosmos Policy / RoboCasa	—	6	144	—	$j = 0, 3, \ldots, 21$
$\pi_{0.5}$ / LP camera L4	5	3	120	$K = 4$	$j = 0, \ldots, 12$
Motus / RoboTwin (replication)	5	3	150	$K = 4$	Main: $j = 0, \ldots, 4$
LingBot-VA / RoboTwin	3	3	108	$K = 4$	Primary: $j = 1, 2$
Astra + scaffold / LIBERO	10	1†	40	$K = 4$	$j = 0, 2, 4, 6, 8$

B SAMPLING POTENTIAL

This appendix reports the complete budget curves, temporal comparisons, and replay controls underlying Section 3.

Protocol and statistical units. For a frozen candidate pool, oracle success O_N is the fraction of roots with at least one successful candidate among the first N; R_N is the mean success of those candidates. Headroom is $H_N = O_N - R_N$. An informative root contains both successful and failed candidates. Candidates are executed from a reset or replayed state, then followed by closed-loop

continuation. RoboTwin uses common continuation random numbers across candidates; Motus commands joint positions, whereas LingBot-VA commands end-effector targets through a planner. Astra includes its waypoint scaffold and stochastic hosted-policy continuations; its API does not honor sampling seeds.

B.1 Complete static candidate-budget curves

Table 5 retains every budget in the main scaling figure and all eight settings in Table 1.

Table 5: Static master curves. I is the informative-root count, except Astra entries marked f, which are fractions (expected fractions for subset rows). $N = 1$ headroom and informative count are zero by definition. Cohort headings give episode/root counts, cluster units, and evaluation seeds or Astra collection labels. All main-table Astra entries use the prefix rows.

N	O_N	R_N	H_N	95% CI(H_N)	I
Astra / LIBERO: 80 roots, 10 tasks, labels 22/23; prefix					
1	—	—	.000	[.000,.000]	0
2	—	—	.206	[.150,.262]	—
4	—	—	.403	[.328,.481]	—
6	—	—	.469	[.383,.542]	—
8	.912	.416	.497	[.422,.573]	$.912^f$
Astra / LIBERO: same 80 roots; all-subset expectation					
1	.416	.416	.000	[.000,.000]	$.000^f$
2	.618	.416	.202	[.159,.234]	$.404^f$
4	.794	.416	.379	[.313,.430]	$.723^f$
6	.870	.416	.454	[.384,.515]	$.848^f$
8	.912	.416	.497	[.422,.573]	$.912^f$
Motus / RoboTwin: 120 episodes, 5 tasks, seeds 1–6					
1	.533	.533	.000	[.000,.000]	0
2	.675	.583	.092	[.054,.129]	22
4	.792	.575	.217	[.144,.331]	53
6	.817	.575	.242	[.178,.336]	59
8	.825	.576	.249	[.193,.334]	66
LingBot-VA / RoboTwin: 120 episodes, 6 tasks, seeds 3–7					
1	.767	.767	.000	[.000,.000]	0
2	.850	.771	.079	[.058,.104]	19
4	.900	.779	.121	[.087,.158]	37
6	.917	.765	.151	[.128,.183]	51
8	.925	.758	.167	[.134,.202]	58
Cosmos Policy / RoboCasa: 384 episodes, 8 tasks, seeds 40–42					
1	.4766	.4766	.0000	[0,0]	0
2	.5156	.4831	.0326	[.0182,.0469]	25
4	.5417	.4831	.0586	[.0326,.0814]	46
6	.5495	.4857	.0638	[.0356,.0911]	53
8	.5547	.4854	.0693	[.0365,.1003]	59
Cosmos Policy / LIBERO-Plus robot L3: 80 episodes, 10 tasks, seeds 0/1					
1	.6250	.6250	.0000	[0,0]	0
2	.6500	.6125	.0375	[.0000,.0875]	6
4	.6500	.6000	.0500	[.0094,.1000]	12
6	—	—	—	—	—
8	.6625	.6031	.0594	[.0172,.1063]	13
Astra / MolmoSpaces: 40 houses/roots, label 22; prefix					
1	—	—	.000	[.000,.000]	0
2	—	—	.088	[.037,.150]	—
4	—	—	.206	[.125,.294]	—
6	—	—	.250	[.154,.346]	—

Table 5: Static master curves (continued).

N	O_N	R_N	H_N	95% CI(H_N)	I
8	.725	.459	.266	[.169,.366]	$.500^f$
Astra / MolmoSpaces: same 40 roots; all-subset expectation					
1	.459	.459	.000	[.000,.000]	$.000^f$
2	.564	.459	.105	[.070,.141]	$.210^f$
4	.652	.459	.192	[.127,.259]	$.377^f$
6	.693	.459	.233	[.151,.316]	$.448^f$
8	.725	.459	.266	[.169,.366]	$.500^f$
GR00T-N1.7 / LIBERO-Plus robot L5: 80 episodes, 10 tasks, seeds 0/1					
1	—	—	0	—	0
2	—	—	.088	[.031,.149]	—
4	—	—	.122	[.063,.182]	—
6	—	—	.142	[.066,.218]	—
8	—	—	.161	[.083,.240]	—
$\pi_{0.5}$ / LIBERO-Plus camera L4: 80 episodes, 5 tasks, seeds 0/1					
1	—	—	0	—	0
2	—	—	.056	[.009,.117]	—
4	—	—	.081	[.015,.166]	—
6	—	—	.108	[.020,.246]	—
8	—	—	.114	[.019,.269]	—
Supplementary RoboCasa breadth: 132 episodes, 6 additional tasks					
4	—	—	.0644	[.0399,.0926]	17
8	—	.1799	.0928	[.0473,.1493]	23
Primary + breadth: 516 episodes, 14 tasks (overlapping analyses)					
4	—	—	.0601	[.0404,.0794]	—
8	—	—	.0753	[.0484,.1044]	82

B.2 Temporal scans and matched comparisons

Table 6: Temporal master results for Table 2. n is the scored count after any stated replay filter; counts of active trajectories before filtering are given in the text/replay table. $\Delta_j = H_4(j) - H_4(0)$ uses the corresponding matched population or, for Astra, each row's own initial-root reference.

Population / replay rule	j	n	H_4	95% CI	Paired Δ_j [95% CI]
Motus / RoboTwin: 120 starts, 5 tasks, seeds 7–9					
Active; exact	0	120	.223	[.163,.281]	—
	1	120	.185	[.119,.250]	—
	2	120	.167	[.106,.250]	—
	4	94	.152	[.075,.211]	—
	8	67	.041	[.009,.065]	—
Matched; exact	0	67	.254	[.164,.370]	Reference
	1	67	.216	[.120,.318]	$-.037$ $[-.193, .086]$
	2	67	.194	[.094,.299]	$-.060$ $[-.228, .068]$
	4	67	.164	[.074,.257]	$-.090$ $[-.264, .053]$
	8	67	.041	[.000,.089]	$-.213$ $[-.353, -.102]$
Cosmos Policy / RoboCasa: 288 starts, 8 tasks, seeds 40–42					
Active; exact	0	288	.0486	[.0252,.0729]	—
	4	288	.0599	[.0330,.0877]	—
	8	288	.0278	[.0156,.0417]	—
	12	254	.0315	[.0119,.0522]	—
Matched; exact	0	254	.0541	[.0279,.0815]	Reference
	4	254	.0669	[.0394,.0956]	—
	8	254	.0315	[.0188,.0444]	—

Table 6: Temporal master results (continued).

Population / replay rule	j	n	H_4	95% CI	Paired Δ_j [95% CI]
	12	254	.0315	[.0119,.0522]	$-.0226$ $[-.0568, .0061]$
Astra / LIBERO: same 20 roots, 10 tasks, initial states 2/3					
All active	4	20	.362	[.250,.462]	$-.006$ $[-.078, .069]$
	8	20	.375	[.238,.512]	$+.007$ $[-.167, .177]$
	12	20	.363	[.225,.488]	$-.006$ $[-.137, .127]$
Drift $< 10^{-6}$	4	17	.368	[.250,.469]	$-.000$ $[-.084, .087]$
	8	14	.357	[.196,.519]	$+.067$ $[-.152, .279]$
	12	13	.269	[.107,.417]	$-.048$ $[-.216, .115]$
LingBot-VA / RoboTwin: 72 starts, 6 tasks, seeds 9–11					
Active; no filter	0	72	.160	[.087,.229]	—
	2	72	.149	[.056,.253]	—
	4	50	.120	[.042,.185]	—
Drift $\leq 10^{-2}$	0	72	.160	[.087,.229]	—
	2	68	.158	[.061,.261]	—
	4	32	.070	[.013,.176]	—
Matched; $\leq 10^{-2}$	0	32	.172	—	Reference
	2	32	.188	—	$+.016$ $[-.116, .134]$
	4	32	.070	—	$-.102$ $[-.224, .000]$
Cosmos Policy / LIBERO-Plus robot L3: 80 starts, seeds 0/1					
Active; no filter	0	80	.0312	[.0062,.0688]	—
	4	80	.0375	[.0000,.0969]	—
	8	38	.0197	[.0000,.0750]	—
	12	32	.0000	[0,0]	—
Matched; $\leq 10^{-4}$	0	30	.033	[0,.110]	Reference
	4	30	.008	[0,.038]	—
	8	30	0	—	—
	12	30	0	—	$-.033$ $[-.107, 0]$
Matched; drift 0	0	13	.077	[0,.250]	Reference
	4	13	.019	[0,.125]	—
	8	13	0	—	—
	12	13	0	—	$-.077$ $[-.250, 0]$

Table 7: Timing of the sampling scans. Environment steps and simulated time are not wall-clock inference costs. RoboTwin's planner uses a variable number of physics substeps, so elapsed simulated time is unavailable.

System	Checkpoints and environment steps	Execution convention
Motus / RoboTwin	$j = 0, 1, 2, 4, 8$; steps $0, 16, 32, 64, 128$	16 actions/chunk; 250 Hz physics; variable planner duration
LingBot-VA / RoboTwin	$j = 0, 2, 4$; steps $0, 48, 112$	First chunk 16 actions, later chunks 32; 250 Hz physics
Cosmos Policy / RoboCasa and LIBERO-Plus	$j = 0, 4, 8, 12$; steps $10, 74, 138, 202$; $0.5, 3.7, 6.9, 10.1$ s	10 settling steps, 16-action chunks; 20 Hz control, 500 Hz physics
Astra / LIBERO	$j = 4, 8, 12$; steps 40, 74–80, 114–120; medians $40, 80, 117$	Up to 10 actions/call; 20 Hz control, 500 Hz physics; median $2.0, 4.0, 5.85$ s
Astra / MolmoSpaces	Static first chunks: 10 steps; temporal scan not tabulated here	66 ms/action; 500 Hz physics; 455-step budget (30 s)

B.3 Replay controls

Repeated execution of the same frozen candidate detects outcome variability without changing its identity. Branch-state replay checks instead test whether alternatives start from the intended

state. These controls characterize the tested executions and states. Table 8 summarizes the controls, including overlapping replay pools.

Table 8: Replay-validity summary.

Policy / environment	Same-candidate execution control	Temporal branch-state replay
Motus / RoboTwin	0/100 outcome flips; 100/100 step-exact (combined controls)	521/521 branches from active trajectories exactly reproduced in the 120-episode scan
LingBot-VA / RoboTwin	9/96 flips, .094 [.050,.169], on six-task controls; 52/96 step-exact	Active: 72/72/50 at $j = 0/2/4$; drift $\leq 10^{-6}$: 72/0/0; $\leq 10^{-2}$: 72/68/32
Cosmos Policy / RoboCasa	No separate flip-rate control available	1,118/1,118 branches from active trajectories exact in the 288-episode temporal scan
Cosmos Policy / LIBERO-Plus	0/160 flips (eight candidates from 20 roots); 20/20 settled states reproduced across processes	111/230 branches exact; 215/230 within 10^{-4}; 15 excluded branches drift up to 1.58
GR00T-N1.7 / LIBERO-Plus	25/160 flips; 10/20 branch states not reproduced across processes	No verified temporal headroom curve available
Astra / LIBERO	Hosted continuations are stochastic; no matched flip-rate control	7/20 roots bit-exact at each j; 17/14/13 meet 10^{-6} at $j = 4/8/12$; other drift up to .89
$\pi_{0.5}$ / LIBERO-Plus	No matched replay control available	No numerical temporal headroom curve available

C SELECTION EFFECTIVENESS

This appendix presents practical-selector results and diagnostic controls for Section 4. Selection gain is $G_N = S_N - R_N$, relative to uniform random selection on the same candidate pool. Gains are success-probability differences, while pairwise accuracy measures whether successful candidates are ranked above failed candidates within a decision state, with ties worth 0.5. Simulator-image judges and classifiers on privileged states are diagnostic controls requiring simulator access. The Motus verifier and Astra diagnostic studies use separate candidate pools from the larger static experiments, with recovery computed only on matched pools.

C.1 PRACTICAL SELECTORS

Action consensus. We evaluate several consensus-style selectors, including proximity to the action mean, medoid selection, k-nearest-neighbor density, an outlier rule, imagined-future consensus, and waypoint consensus. We additionally report a label-majority diagnostic that selects from the majority outcome class and serves as an oracle-label reference. The dedicated LingBot follow-up uses action-mean consensus.

Table 9: Consensus gains over random selection. Brackets are 95% intervals; a dash denotes an unavailable estimate. The first three rows summarize exploratory consensus studies.

Cohort	Oracle H_N	Label-majority gain	Consensus gain
Motus / RoboTwin; balanced; $N = 8$; episode count unreported	$+.269$	$+.010$	$+.002$
LingBot / RoboTwin; early majority-success; 3 tasks; N and count unreported	$+.156$	$+.073$	$+.073$; CI crosses zero, endpoints unreported
Astra / LIBERO; minority-success; proposal identity, N, count unreported	$+.463$	$-.088$	$+.013$
LingBot / RoboTwin; 48 episodes; 6 tasks; seeds 3,4; $N = 8$	—	$+.068$ $[.016, .120]$	$+.005$ $[-.068, .076]$

Motus image verifiers. Table 10 evaluates image-based candidate verification on the 60-episode Motus / RoboTwin pool using the matched oracle reference. Configuration 2 scores root/future-frame

pairs with a graded candidate-approach rubric. Astra configuration 2 yields a positive gain at $N = 2$ ($+.025$ $[.008, .043]$), while its intervals at $N = 4$ and $N = 8$ include zero. Its point-estimate oracle-headroom capture decreases from 25% at $N = 2$ to 11% at $N = 8$, showing that the verifier recovers only a limited fraction of the available scaling headroom.

Table 10: Motus verifier gains on 60 RoboTwin episodes, five tasks, seeds 1–3. Brackets: 95% CIs. Capture is the point-estimate fraction of this pool's oracle gain.

Selector	$N = 2$	$N = 4$	$N = 8$
Oracle	$+.100$	$+.196$	$+.269$
	$[.050, .150]$	$[.113, .321]$	$[.202, .360]$
Astra configuration 1	$-.050$	$-.054$	$-.048$
Astra configuration 2	$+.025$	$+.038$	$+.030$
	$[.008, .043]$	$[-.004, .085]$	$[-.021, .088]$
Gemma configuration 2	$-.017$	$-.071$	$+.002$
Astra configuration 2 capture	25%	19%	11%

Cosmos Policy native values and VLM judging. The RoboCasa comparison ranks a frozen eight-candidate pool at the initial decision state and measures success after each candidate's closed-loop continuation. Native values come from the same checkpoint as the proposals. This controlled comparison uses our benchmark's candidate pools and selection protocol, which differ from the published Cosmos Policy evaluation.

Table 11: Cosmos Policy / RoboCasa on the same 384-episode, $N = 8$ pool as the primary static study (eight task clusters). Gain intervals are paired 95% CIs. The last column is success on 59 informative roots, not pairwise accuracy.

Selection rule	Success	Gain [95% CI]	Informative-root success
Oracle	.5547	$+.0693$ $[.0365, .1003]$	1.000 (definition)
Uniform random	.4854	Reference	.5487 (expectation)
Native value head	.4714	$-.0140$ $[-.0352, .0046]$	.4576 (27/59)
Gemma-4-31B-IT	.4766	$-.0088$ $[-.0221, .0052]$	.4915 (29/59)
First candidate	.4766	$-.0088$ $[-.0208, .0046]$	.4915 (29/59)

C.2 Predicted, simulator-image, and privileged-state controls

Table 12: Initial-branch image/state diagnostic cohorts. Inf.: informative roots; pairs: discordant candidate pairs. Pools and task clusters are counted separately.

Policy / condition	Pools; N	Roots	Inf.	Pairs	Clusters
Motus / RoboTwin; native WM	3; 8	60	34	415	5
GR00T / LIBERO-Plus robot L5/L4/L3; Cosmos-Policy-WM	4; 4	112	30	93–96	10
GR00T / same candidates; corrected WAP	same; 4	112	29	93	10
Astra waypoint / LIBERO spatial; labels 20,21	2; 4	40	29	96	10
$\pi_{0.5}$ / LIBERO-Plus camera L4/L3	2; 4	80	22	67	8
LingBot / RoboTwin	2; 8	48	22	236	6

Judges and statistical evaluation. Gemma-4-31B-IT (temperature 0) and GPT-6 Astra (temperature unspecified) see root/future images and a fixed 0–100 candidate-approach rubric; their scores are never pooled. Astra judges its own candidates in the Astra-policy cohort. Pairwise evaluation uses task-clustered percentile-bootstrap intervals and within-root permutation tests; $\star$ marks reported $p < .05$, including inverted rankings. Paired contrasts use the intersection of scored roots. Argmax gains use exact expected success under ties and clustered paired bootstraps. Between-method comparisons use paired contrasts; tests against chance assess each evaluator separately. Initially empty API

responses were re-scored until all analyzed responses parsed successfully; initial missing-response counts are unavailable.

Table 13: Initial-branch image-judge accuracies for Table 3B and Figure 2. Imagined denotes one WM call; simulator entries give the reported horizons for each judge. Chance is .5. Pointwise CIs are unavailable; $\star$: permutation $p < .05$.

Policy / WM	Judge	Imagined	Simulator image: horizon, accuracy
Motus / native WM	Gemma	.534	$h1$: .523; $h4$: .583*; $h8$: .605*
	Astra	.521	$h1$: .606*; $h2$: .467; $h4$: .590*; $h8$: .746*
GR00T / Cosmos-Policy-WM	Gemma	.521	$t16$: .458; $t64$: .589; $t256$: .276*
	Astra	.542	$t16$: .484; $t64$: .625*; $t128$: .719*; $t256$: .797*
GR00T / corrected WAP	Gemma	.495	Same candidate simulator-image controls
	Astra	.597	Same candidate simulator-image controls
Astra waypoint / Cosmos-Policy-WM	Gemma	.505	$t10$: .505; $t16$: .458; $t40$: .52 (label 21 only)
	Astra (self)	.359*	$t10$: .578; $t16$: .396; $t40$: .46 (label 21 only)
$\pi_{0.5}$ / external WM	Gemma	.500	$t10$: .448; $t40$: .485
	Astra	.604	$t10$: .470; $t40$: .470
LingBot / native WM	Gemma	Unmeasured	$h1$: .521; $h2$: .525; $h4$: .513
	Astra	Unmeasured	$h1$: .405; $h2$: .542; $h4$: .581

The $\pi_{0.5}$ pool contains Cosmos-Policy-WM and WAP predictions, but its aggregate scores cannot be attributed separately to the two WMs. Motus Gemma at $h2$ is unavailable. Simulator-image discrimination appears at several horizons, including Motus Astra at $h1$ and Gemma at $h4, h8$. Long-horizon images may reveal task completion and therefore measure retrospective outcome recognition as well as prediction. WAP results use corrected upright predictions and exclude the earlier vertically inverted judge and perception controls. The corrected paired WAP–Cosmos intervals span zero.

Privileged-state classifiers. A linear pairwise head uses root-centered state features, is fit per base task, and is evaluated leave-one-root-out. Nested evaluation selects regularization inside each outer training fold; same-fold selection uses the scored folds and can be optimistic. These classifiers probe state information under a particular supervised evaluator; comparisons with image judges also reflect differences between evaluators.

Table 14: Initial-branch state-classifier results for Table 3C and Figure 2. Action-only heads are separate supervised diagnostics. Pointwise intervals are unavailable; $\star$: $p < .05$; n.s.: no significant result reported.

Policy / simulator	True state: horizon, accuracy	Action only	State evaluation
Motus / RoboTwin	$h1$: .66*; $h2$: .61*; $h4$: .59*; $h8$: .80*	.52	Nested
GR00T / LIBERO-Plus	$t8$: .22*; $t32$: .57; $t64$: .68*; $t128$: .67*; $t256$: .90*	.37	Nested; $t8$ inverted
Astra waypoint / LIBERO	$t16$: .42; $t40$: .68 (n.s.)	.62	Label 21 only; nesting unspecified
$\pi_{0.5}$ / LIBERO-Plus	$t5$–$t80$: .49–.58 (n.s.); individual horizons unreported	.42	Nested
LingBot / RoboTwin	$h4$: .70/.71 for the two seeds	—	Same-fold only; nested unstable

Motus's state accuracy varies nonmonotonically before $h8$; GR00T's first reported positive state result is at $t64$. The $\pi_{0.5}$ range summarizes multiple horizons; neither it nor Astra's .68 is reported significant. LingBot has same-fold estimates and unstable nested evaluation.

Table 15: Paired image contrasts (A) and actual selection gains (B), with distinct metrics. C: simulator image; D: single-call imagined image. A uses shared scored roots; its last column counts roots favoring the first/second input (others may tie). B selects the highest score on each policy's own candidate pool. Brackets: clustered paired 95% CIs.

A. Paired differences in pairwise accuracy

Contrast	Difference [95% CI]	Roots first/second
Motus / Astra: $C_{h1} - D$	$+.09$ $[-.08, .33]$	$20/13$
Motus / Astra: $C_{h8} - D$	$+.23$ $[.05, .35]$	$25/6$
Motus / Gemma: $C_{h8} - D$	$+.07$ $[-.09, .19]$	$19/14$
GR00T / Astra: $C_{t16} - D$	$-.06$ $[-.19, .08]$	$11/13$
GR00T / Astra: $C_{t256} - D$	$+.26$ $[.10, .43]$	$19/5$
GR00T / Gemma: $D_{\mathrm{WAP}} - D_{\mathrm{Cosmos}}$	$-.04$ $[-.18, .06]$	—
GR00T / Astra: $D_{\mathrm{WAP}} - D_{\mathrm{Cosmos}}$	$+.07$ $[-.07, .25]$	—
Astra self-judge: $C_{t10} - D$	$+.22$ $[.03, .37]$	$20/9$
Astra self-judge: $C_{t16} - D$	$+.04$ $[-.13, .20]$	$14/10$
$\pi_{0.5}$ / Astra: $C_{t10} - D$	$-.13$ $[-.30, .05]$	$5/13$

B. Success-probability gains over random selection

Policy / judge; simulator horizon	Gain [95% CI]	N
Motus / Astra; $h8$	$+.138$ $[-.001, .301]$	8
Motus / Gemma; $h8$	$+.075$ $[-.002, .161]$	8
GR00T / Astra; $t256$	$+.069$ $[.030, .118]$	4
GR00T / Astra; $t64$	$+.027$ $[.014, .042]$	4

Paired contrasts are computed on shared roots, so they can differ from differences of rounded marginal accuracies. The contrasts $C_{h8} - D$ and $C_{t256} - D$ compare long simulator futures with *short imagined* futures. Imagined-input gain aggregates range from $-.10$ to $+.06$, with all available intervals including zero; per-policy estimates and interval endpoints are unavailable.

C.3 Long-horizon recursive Motus rollout

At the first call, the protocol uses the frozen action chunk and Motus's predicted frame. At calls 2–8, the previous predicted composite image is fed back as the observation, and the last commanded joint target supplies approximate proprioception; each call uses the corresponding simulator tail's policy seed. The experiment covers 480 candidates (60 roots, $N = 8$), with 3,360 additional passes and zero execution errors. This open-loop recursive-input test can accumulate error from predicted observations and approximate proprioception; its results characterize this rollout protocol.

Table 16: Motus pairwise accuracy under recursive imagination versus simulator images, underlying Figure 2. h counts 16-step lookahead units. Brackets: available pointwise 95% intervals; simulator intervals and Gemma's simulator $h2$ estimate are unavailable.

h	Imagined: Gemma	Imagined: Astra	Simulator: Gemma	Simulator: Astra
1	.534	.521	.523	.606
2	.525 [.46, .60]	.477 [.41, .56]	—	.467
4	.518 [.45, .58]	.522 [.41, .62]	.583	.590
8	.510 [.40, .60]	.554 [.48, .62]	.605	.746

At $h8$, the paired imagined-minus-simulator difference is $-.10$ $[-.16, -.003]$ for Gemma and $-.19$ $[-.30, -.01]$ for Astra; 25/34 informative roots favor simulator images for Astra. Selection gains against random selection are a separate metric (Table 15B). As a descriptive check, own-candidate identification in DINOv2 CLS space is .148/.156/.158/.117 at $h = 1/2/4/8$ (chance $1/8 = .125$), while distance to the corresponding true future is .56/.73/.84/.96. These descriptive checks have no intervals or tests and cannot distinguish the sources of rollout error.

Scope and limitations. Few task clusters limit precision, and intervals containing zero remain compatible with both gains and losses. Some paired comparisons lack reported uncertainty intervals. Astra state nesting and action-head regularization selection are unspecified. A cohort-wide replay rate is unavailable for the mixed-level GR00T selector pool. The experiments evaluate the stated selectors, excluding value heads trained on imagined futures. Differences in cohorts, evaluators, and compute constrain cross-model comparisons.

D UNCERTAINTY INFORMATIVENESS

D.1 PROTOCOL AND COMPLETE MAIN-FIGURE RESULTS

Action dispersion is the mean coordinate-wise standard deviation across action sequences sampled from the same observation, averaged over their common time prefix. LIBERO-Plus, RoboTwin, and Astra monitoring use $K = 4$ auxiliary draws that are used only to measure dispersion, not to select the executed action. Unless stated otherwise, higher scores predict eventual *failure*, and AUROC 0.5 denotes chance ranking.

Confidence intervals generally use 2,000 percentile bootstrap resamples over task or base-task clusters. Two-sided within-task permutation tests use a plus-one correction, with 5,000 permutations for the pooled LIBERO-Plus and Astra analyses and 2,000 for RoboTwin and sensitivity analyses. The Astra initial-root analysis uses 5,000 bootstrap resamples. Five-cluster and three-cluster analyses naturally yield wider or more discrete uncertainty estimates.

Table 17 gives all 41 points in Figure 3. Environment steps specify the timing of each checkpoint: Cosmos Policy uses $10 + 16j$, $\pi_{0.5}$ uses $10 + 10j$, and Motus uses $16j$. Astra monitoring $j = 0$ follows the first executed chunk. Cosmos breadth and Astra use fixed cohorts; the other curves use episodes still active at each checkpoint, so later estimates are conditional on remaining active.

Table 17: Complete numerical counterpart of Figure 3. Active counts the episodes in the checkpoint cohort; F/S denotes failures/successes. Intervals are 95% intervals, and p denotes the within-task permutation result.

j	Env. steps	Active	F/S	AUROC	95% CI	p
Cosmos Policy / LIBERO-Plus spatial L3: 375 episodes, 5 seeds, 10 clusters						
0	10	375	169/206	.6448	[.541, .789]	.0002
1	26	375	169/206	.6583	[.501, .778]	.0002
2	42	375	169/206	.6162	[.465, .741]	.0410
3	58	375	169/206	.6548	[.567, .726]	.0002
4	74	375	169/206	.8322	[.740, .916]	.0002
5	90	370	169/201	.8920	[.798, .960]	.0002
Cosmos Policy / three additional LIBERO-Plus suites: fixed 295, 1 seed, 30 clusters						
0	10	295	118/177	.625	[.548, .695]	.016
2	42	295	118/177	.782	[.727, .829]	.0002
4	74	295	118/177	.780	[.708, .846]	.0002
5	90	295	118/177	.753	[.669, .826]	.0002
$\pi_{0.5}$ **/ LIBERO-Plus camera L4: 120 episodes, 3 seeds, 5 clusters**						
0	10	120	28/92	.811	[.390, .882]	.036
1	20	120	28/92	.815	[.520, .855]	.205
2	30	120	28/92	.816	[.589, .871]	.038
3	40	120	28/92	.704	[.461, .739]	.698
4	50	120	28/92	.550	[.438, .775]	.317
5	60	120	28/92	.519	[.380, .631]	.765
6	70	120	28/92	.650	[.499, .754]	.050
7	80	120	28/92	.715	[.578, .758]	.016
8	90	120	28/92	.800	[.619, .847]	.0028
9	100	120	28/92	.719	[.369, .809]	.351
10	110	119	28/91	.684	[.396, .742]	.615
11	120	110	28/82	.701	[.540, .774]	.102

j	Env. steps	Active	F/S	AUROC	95% CI	p
12	130	93	28/65	.517	[.419, .588]	.743
Motus / RoboTwin: 150 episodes, 3 held-out seeds, 5 clusters						
0	0	150	64/86	.458	[.383, .550]	.33
1	16	150	64/86	.540	[.494, .605]	.44
2	32	150	64/86	.518	[.422, .673]	.75
3	48	132	64/68	.599	[.475, .749]	.020
4	64	110	64/46	.564	[.421, .702]	.46
Cosmos Policy / RoboCasa: 144 episodes, 6 reported seeds						
0	10	144	94/50	.624	[.560, .700]	.006
3	58	144	94/50	.547	[.472, .604]	.31
6	106	144	94/50	.520	[.424, .628]	.58
9	154	144	94/50	.641	[.517, .787]	.002
12	202	143	94/49	.605	[.506, .733]	.037
15	250	133	94/39	.639	[.512, .767]	.014
18	298	122	94/28	.629	[.496, .735]	.11
21	346	116	94/22	.584	[.486, .671]	.20
Astra + scaffold / LIBERO: fixed 40 episodes, 1 collection group, 10 clusters						
0	10	40	26/14	.283	[.128, .566]	.187
2	30	40	26/14	.497	[.311, .678]	.971
4	48–50	40	26/14	.772	[.642, .910]	.008
6	66–70	40	26/14	.652	[.515, .805]	.076
8	85–90	40	26/14	.538	[.363, .714]	.625

D.2 Cosmos Policy: seed replication and task-family breadth

The spatial robot-initial-state L3 study evaluates 75 episodes per seed with $K = 4$ sampled action chunks. Seed 0 is the discovery seed and seeds 1–4 provide independent replication. Action dispersion is strongly predictive at the two later landmarks: all five seed-level estimates at $j = 4$ and $j = 5$ exceed chance, with pooled AUROC .8322 $[.740, .916]$ and .8920 $[.798, .960]$, respectively. The rise through $j = 4$ occurs before cohort attrition; five successful episodes terminate before $j = 5$. The pooled $j = 0$ interval also excludes chance, indicating that the signal is not restricted to late execution.

Table 18: Cosmos Policy seed replication and task-family breadth at the two primary landmarks. Spatial F/S counts refer to each full 75-episode seed; breadth results use the fixed cohort. Brackets are 95% CIs.

Seed / task suite	F/S	$j = 4$ AUROC [CI]	$j = 5$ AUROC [CI]
Spatial L3: 10 base-task clusters			
Seed 0 (discovery)	33/42	.823 [.741, .923]	.880 [.768, .970]
Seed 1	32/43	.823 [.702, .926]	.902 [.815, .958]
Seed 2	35/40	.852 [.756, .930]	.892 [.778, .960]
Seed 3	36/39	.843 [.723, .940]	.900 [.799, .963]
Seed 4	33/42	.820 [.727, .911]	.892 [.798, .966]
Pooled	169/206	.8322 [.740, .916]	.8920 [.798, .960]
Three-suite extension: 10 base tasks per suite			
Object ($n = 90$)	35/55	.851 [.796, .913]	.842 [.733, .937]
Goal ($n = 105$)	55/50	.782 [.602, .922]	.663 [.531, .783]
LIBERO-10 ($n = 100$)	28/72	.697 [.567, .799]	.728 [.568, .871]
Pooled ($n = 295$)	118/177	.780 [.708, .846]	.753 [.669, .826]

Task-family breadth. The same protocol extends without retuning to LIBERO object, goal, and LIBERO-10 at robot L3. The fixed cohort contains 295 episodes (118 failures and 177 successes) across 30 task clusters. The pooled AUROC remains high at both later landmarks, reaching .780 $[.708, .846]$ at $j = 4$ and .753 $[.669, .826]$ at $j = 5$, with permutation $p = .0002$ at both

checkpoints. Pairs-weighted within-suite AUROCs are similarly .776 and .734, indicating that the pooled result is not driven by a single task family.

D.3 ASTRA: FIXED-COHORT MONITORING AND ROOT-LEVEL CONTROLS

We evaluate action dispersion for Astra on a fixed cohort of 40 LIBERO trajectories spanning ten spatial tasks. At each monitored checkpoint, $K = 4$ passive action samples are drawn from the current observation and used only to measure dispersion. All 40 trajectories (26 failures and 14 successes) remain active throughout the monitoring window, so the checkpoint comparisons are not affected by survivor attrition. Monitoring $j = 0$ follows the first executed chunk (step 10); subsequent checkpoints are $j = 2, 4, 6, 8$.

Table 19: Astra fixed-cohort monitoring and separate initial-root controls. Monitoring rows use $K = 4$ passive draws; p is the within-task permutation test, p_P is the pooled test, and Z counts zero-spread episodes. Root-level rows report dispersion over candidate pools at the initial state.

Study / signal	j or K	AUROC	95% CI	p	p_P	Z
Passive monitoring: fixed 40 episodes, 10 task clusters						
Action (j)	0	.283	[.128, .566]	.187	.024	12/40
	2	.497	[.311, .678]	.971	.982	2/40
	4	.772	[.642, .910]	.008	.004	4/40
	6	.652	[.515, .805]	.076	.117	6/40
	8	.538	[.363, .714]	.625	.695	12/40
LIBERO initial roots: 80 roots, 43/37 F/S, 10 clusters						
Action (K)	8	.355	[.214, .554]	.026	—	—
Action	4	.383	[.233, .585]	.069	—	—
Waypoint	8	.466	[.312, .616]	.60	—	—
Waypoint	4	.549	[.373, .734]	.44	—	—
MolmoSpaces initial roots: 40 houses, 21/19 F/S						
Action (K)	8	.414	[.220, .601]	.36	—	—
Action	4	.401	[.221, .586]	.29	—	—
Waypoint	8	.586	[.406, .768]	.37	—	—
Waypoint	4	.604	[.420, .780]	.26	—	—

Fixed-cohort monitoring. Action dispersion is most informative at the intermediate checkpoint $j = 4$ (48–50 environment steps, 2.4–2.5 seconds), where failure-prediction AUROC reaches .772 [.642, .910] with within-task $p = .008$. The $j = 6$ estimate is also above chance (.652 [.515, .805]) but is less conclusive under the within-task test ($p = .076$). The remaining monitored checkpoints do not show clear positive failure-prediction signals. Because the same 40 trajectories are evaluated at every checkpoint, these differences are not driven by changes in cohort membership.

Initial-root controls. A separate root-level analysis evaluates dispersion over candidate pools before trajectory monitoring. On LIBERO, action-dispersion AUROCs are .355 [.214, .554] for $K = 8$ and .383 [.233, .585] for $K = 4$; waypoint dispersion is .466 [.312, .616] and .549 [.373, .734], respectively. MolmoSpaces shows similarly inconclusive root-level estimates. These controls therefore do not provide a clear initial-state failure signal, in contrast to the intermediate-checkpoint signal observed in fixed-cohort monitoring.

D.4 VALUE AND IMAGINED-FUTURE CONTROLS

Table 20 compares value dispersion and imagined-future dispersion with action dispersion. Value SD is scored higher→failure. Imagined-future dispersion uses decoded final frames jointly generated with different sampled actions. Residual signals are computed by within-task rank-regression against action dispersion.

Table 20: Value and imagined-future controls. AUROC [95% CI] predicts eventual failure; dashes indicate unavailable tests.

Cohort / checkpoint	Signal	AUROC [95% CI]	p
Spatial, $j = 4$, $n = 375$	Value SD	.831 [.712, .932]	—
Spatial, $j = 5$, $n = 370$	Value SD	.819 [.706, .907]	—
Spatial, $j = 5$	Action residual on value SD	.754 [.640, .847]	—
Spatial, $j = 5$	Value-SD residual on action	.575 [.473, .683]	—
Breadth, fixed $n = 295$, $j = 0$	Value SD	.587 [.498, .670]	.016
Breadth, $j = 2$	Value SD	.738 [.676, .795]	.0002
Breadth, $j = 4$	Value SD	.717 [.623, .810]	.0002
Breadth, $j = 5$	Value SD	.744 [.667, .809]	.0002
Motus, $j = 2$, $n = 150$	Imagined final frame	.662 [.540, .743]	.040
Motus, $j = 2$	Imagined residual on action	.557 [.491, .623]	—
LingBot-VA, $j = 2$, $n = 108$	Imagined final frame	.692 [.608, .852]	.024
LingBot-VA, $j = 2$	Imagined residual on action	.642 [.581, .819]	—

Value signals. For Cosmos Policy, value dispersion is informative at later checkpoints. In the spatial-task study, value-SD AUROC reaches .831 [.712, .932] at $j = 4$ and .819 [.706, .907] at $j = 5$. The three-suite breadth study shows similarly positive signals, with AUROC .738, .717, and .744 at $j = 2, 4, 5$, respectively. However, action and value dispersion are strongly correlated ($\rho = .757$ at $j = 4$ and .756 at $j = 5$). At $j = 5$, action dispersion residualized on value remains predictive (.754 [.640, .847]), whereas value dispersion residualized on action does not (.575 [.473, .683]). Thus, value dispersion provides a useful secondary signal but no established incremental benefit beyond action dispersion.

Imagined-future signals. For Motus, imagined final-frame dispersion predicts failure at $j = 2$ (.662 [.540, .743]), but the residual after controlling for action dispersion is .557 [.491, .623], providing no clear additional signal. LingBot-VA shows stronger secondary evidence: its $j = 2$ final-frame dispersion reaches .692 [.608, .852], and the residual remains above chance at .642 [.581, .819]. Because this analysis covers only three task clusters, we treat the LingBot-VA residual as exploratory.

RoboCasa whole-episode value signal. A separate RoboCasa whole-episode study tests whether value mean predicts success across runs. Mean run-level AUROC is .615 across eight discovery runs (336 episodes; range .537–.706), but held-out seeds 30/31/32 yield .403/.500/.484 (72 episodes; mean .463), and seeds 3–10 average .482 (192 episodes; range .127–.659). Thus, the discovery relationship does not reproduce in either held-out block.

Overall, value and imagined-future signals can contain failure information, but additional predictive information beyond action dispersion is setting-dependent.