REPEAT BEFORE YOU ROUTE: EVALUATING ADAPTIVE COMPUTE FOR VLA POLICIES

Anonymous authors
Paper under double-blind review

ABSTRACT

Adaptive robot compute depends on a simple question: which tasks benefit from a different compute setting? We introduce REPEAT-TO-SELECT, a method that measures the opportunity for this choice using repeated rollouts. A repeat-product estimator isolates stable differences in success probability; unbiased centering then separates the advantage of a better default from task-dependent selection opportunity. We derive a sharp upper bound on the gain available to a task-based selector and a local lower bound linking the number of tasks, repeats, and detectable effect size. Together, these results turn repeated outcomes into a procedure for choosing defaults, measuring selection opportunity, and allocating an evaluation budget.

On 120 tasks with three repeats per setting, increasing the executed action chunk from one to ten improves success by 5.83 percentage points and reduces policy calls by 91.56%. A separate 360-episode five-action comparison supports ten-action execution with 50.17% fewer calls and a nominal success-loss bound within two points. Solver-depth selection remains statistically unresolved: its nominal ceiling is 4.17 points and its conservative ceiling is 26.72 points. Across 110,610 public code and agent outcomes, apparent oracle gains exceed cross-validated same-task selection gains in all eight panels. The method and released outcomes provide a common basis for improving fixed settings and evaluating task-based compute allocation.

1 INTRODUCTION

A robot placing a bowl on a plate has two ways to spend more time on its policy. It can refine the next action plan for longer, or it can stop more often to look again and replan. These choices affect different parts of the control loop, creating a trade-off between plan refinement and feedback frequency. In our robot evaluation, executing a longer chunk before replanning improves success while sharply reducing policy calls. Finding that better setting is already useful, even before building an adaptive controller.

The harder question is whether different tasks benefit from different settings. Imagine that a cheap setting succeeds on one task while an expensive setting fails, with the reverse happening on another task. Those outcomes look like training labels for a selector. Yet small changes in a grasp or observation can also change the outcome of the same setting. We need a way to separate useful differences from this ordinary repeat variation (Figure 1).

We introduce REPEAT-TO-SELECT, a method for making that separation. For each task, it compares outcomes across settings and across independent repeats of the same setting. Subtracting the latter isolates the squared difference in success probability. We then remove the average advantage of each setting: a setting that is consistently better contributes to the default; changes in its relative advantage across tasks contribute to selection opportunity. The remaining variation gives an upper bound on what task-based selection could add.

The distinction between opportunity and achieved selection is central. Two independent attempts that each succeed with probability 50% succeed at least once with probability 75%. That is a useful best-of-two outcome when both attempts can be run and verified. A compute selector instead chooses a setting before execution, using the task information available at that point. REPEAT-TO-SELECT

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.

Repeat the same task

	Attempt 1	Attempt 2
Setting A	success	*failure*
Setting B	*failure*	success

Which differences persist across attempts?

Illustrative outcomes, not recorded trials.

Figure 1: Repeated attempts separate realized winners from stable setting effects. The hypothetical outcomes illustrate how the winner can change between attempts. The protocol uses independent repeats to estimate the stable component before assessing task-based selection.

estimates the stable differences relevant to this choice; held-out evaluation measures how much of them a learned selector captures.

Our main contributions are:

1. **A repeat-based estimator of selection opportunity.** We formulate compute selection in terms of stable, task-specific success probabilities. Building on the squared-distance identity from energy statistics, we derive a repeat-product estimator and an unbiased finite-panel centering correction. These separate average setting gains from the task-dependent effects that a selector could exploit.
2. **A sharp selection bound and a repeat-budget lower bound.** We translate centered setting effects into a sharp upper bound on the gain of any task-based selector. A local lower bound characterizes how task count, repeats, and effect size jointly determine detection difficulty. The two results connect the value of adaptation to the evidence needed to measure it.
3. **An executable procedure for compute allocation.** We combine fixed-setting comparisons, corrected opportunity estimates, and held-out selector evaluation in one workflow. On 120 robot tasks, ten-action execution improves success by 5.83 percentage points over one-action execution while reducing policy calls by 91.56%. A further 360-episode comparison with five-action execution yields 50.17% fewer calls and passes the nominal two-point success-loss screen. We release the repeated outcomes and implementation, with applications to six earlier robot panels and eight public code/agent panels.

Section 3 develops the estimator and guarantees, followed by their application to robot compute and public outcome panels.

2 Related Work

Adaptive inference beyond robotics. Model-routing benchmarks expose prompt-by-model outcome matrices and compare quality–cost selectors (Hu et al., 2024). Best-of-N and pass@k methods instead spend multiple samples on one prompt and select or aggregate after generation (Chen et al., 2021; Huang et al., 2025). Both use maxima over stochastic outcomes, but their deployment semantics differ. Running all samples makes best-of-N a real compute strategy; using the same maximum to claim that a pre-outcome router can choose a single winner creates the winner's-curse problem studied here. Our estimand applies whenever public artifacts contain at least two independent outcomes per instance–arm cell; single-output routing matrices lack this unbiased correction.

Efficient VLA inference. π_0 and $\pi_{0.5}$ generate action chunks by integrating a learned flow (Black et al., 2024; Physical Intelligence, 2025). ProbeFlow adapts integration effort from a solver-side signal (Fang et al., 2026). AtVLA gates high-resolution visual refinement from action disagreement

(Cui et al., 2026), while EMS switches between decoupled policy scales using environment feedback (Sun et al., 2026). Token pruning, caching, and distilled generators reduce other parts of the inference path, while OpenVLA provides an autoregressive architecture-family contrast (Kim et al., 2025). These approaches ask how to allocate compute. We ask a logically prior question for task-only menus: whether observed binary arm differences contain stable task-level opportunity before a router is trained.

Adaptive chunk execution. A^3, AAC, AutoHorizon, PACE, and DVAC select horizons from predicted actions, attention, kinematics, or denoising dynamics (Chen et al., 2026; Liang et al., 2026; Wang et al., 2026a; Nie et al., 2026; Feng et al., 2026). BCP learns continue-or-replan decisions from trajectory reward (Xu et al., 2026); FFDC checks imagined futures against real observations for world action models (Wang et al., 2026b); VLA-Corrector and SV-VLA trigger replanning from online visual evidence (Pan et al., 2026; Wang et al., 2026c); RTC and FutureRTC condition generation around execution-time commitments (Black et al., 2025; Jiang et al., 2026). These methods can improve control even when task identity is uninformative because they use within-episode state, phase, uncertainty, or latency. Our bound deliberately covers task-only routing among a fixed set of settings. Richer information sets require matching at their decision context. The controller comparison includes A^3 as a descriptive external controller. Because it changes decisions within an episode, we evaluate it separately as an online controller.

Evaluation noise, psychometrics, and repeated evaluation. Benchmark-variance work asks whether score differences are statistically meaningful (Bouthillier et al., 2021; Madaan et al., 2024; Miller, 2024). Recent LLM evaluations show that multiple generations reveal prompt-level stochasticity hidden by a single output (Song et al., 2024; Zhang et al., 2025); repeated SWE-bench runs expose the same issue in software agents (Bjarnason et al., 2026). Item-response theory models item difficulty and discrimination (Lalor et al., 2016); it is complementary to our nonparametric identification question, which asks whether two arms' instance-wise probabilities differ before imposing a latent ability model. Repeated robot rollouts and multilingual code-generation suites such as MultiPL-E (Cassano et al., 2023) provide natural repeat panels when their sampling provenance is retained. Repeats are normally used to shrink confidence intervals on mean success. Here repeats play a different role: their disagreement is a measurement-error control for the *labels used to justify adaptation*. With squared distances, the cancellation is the exponent-two endpoint of the distance-components/energy-statistics decomposition (Székely & Rizzo, 2013). Our contribution is its task-local use as an arm effect estimand, its unbiased finite-panel centering, the sharp routing ceiling, and an outcome-independent VLA protocol that makes the assumptions auditable.

3 A Repeat-Based Method for Choosing Compute

Repeat-to-Select takes a table of binary outcomes: task, compute setting, repeat, and success. It returns three complementary results: average performance and compute for each setting, corrected differences between settings, and an upper limit on the gain from task-based selection. We use *arm* as shorthand for a setting and *repeat* for another rollout on the same task and nominal initial state.

What information the selector receives. The reported selection bounds apply to task identity, averaging over observations that vary between repeats. Recorded initial features differ on 77 of the 120 tasks, so identical nominal initial-state indices do not establish identical camera inputs. Image-conditioned and within-episode routers require repeats matched at their own decision context. This scope applies to every reported selection ceiling below.

Learning objective. Let $p_{t,a}$ be the success probability of setting a on task t, and let $\mu_a = \mathbb{E}_T p_{T,a}$ average over the evaluated task panel. A task-conditioned selector g chooses one of K settings before execution. Its improvement over the best fixed setting is

$$\Delta(g) = \mathbb{E}_T p_{T,g(T)} - \max_a \mu_a.$$

We measure the opportunity for positive $\Delta(g)$ before fitting g. This separates the existence of useful task-dependent choices from the ability to learn those choices on held-out tasks.

3.1 MATCH TASKS ACROSS INDEPENDENT REPEATS

Choose the settings and task panel before inspecting their outcomes. Run each setting at least twice per task, keeping the policy and outcome definition fixed and using independent repeat randomness. More repeats improve precision; the precision depends on the number of tasks and effect size as well. Our robot evaluation uses three.

Repeats must have a stable distribution within a task and setting. Different repeat labels must be independent; outcomes sharing a seed can be coupled across settings. We therefore compare different labels throughout. With three repeats, this gives three within-setting pairs and six cross-setting pairs per task and setting pair. Tasks are the independent sampling units.

3.2 ESTIMATE STABLE SETTING EFFECTS

Two settings can produce different outcomes because their success probabilities differ, or simply because either rollout can succeed or fail. Within-setting disagreement measures the second source. Write $C_{t,ab}$ for disagreement between settings a, b on different repeats, and $R_{t,a}$ for disagreement between two independent repeats of a. Then

$$\underbrace{H_{t,ab}}_{\text{corrected effect}} = C_{t,ab} - \frac{R_{t,a} + R_{t,b}}{2} = (p_{t,a} - p_{t,b})^2.$$

The equality follows by expanding the Bernoulli disagreement probabilities: the two within-setting variance terms cancel. Thus subtracting repeat noise isolates a stable difference in success probability. Averaging the observed pair comparisons gives an unbiased estimate. A positive population value means the settings differ; mean success supplies the direction of that difference. Finite-data estimates can be negative. Retaining their signed values preserves the estimator's unbiasedness and makes its sampling variation visible.

Repeats supply the second-order information needed for this estimate. Independent observations from the same cell make the squared probability terms directly estimable. The resulting guarantee holds over arbitrary cell probabilities; structured pooling and Bayesian models offer complementary ways to incorporate additional information.

3.3 ISOLATE TASK-DEPENDENT SELECTION OPPORTUNITY

Suppose setting A is ten percentage points better than B on every task. Choosing A everywhere captures the entire gain. A selector has something left to exploit only when the relative benefit changes across tasks. We therefore remove each setting pair's average effect and measure the variation that remains, called *centered dispersion* (V). Writing $\delta_{t,ab} = p_{t,a} - p_{t,b}$ makes this explicit:

$$V = \binom{K}{2}^{-1} \sum_{a<b} \mathbb{E}_T\left[(\delta_{T,ab} - \mathbb{E}_T \delta_{T,ab})^2\right].$$

Simply subtracting the square of an estimated average would introduce another bias. Our estimator instead uses products of effect estimates from different tasks, whose errors are independent. This produces an unbiased finite-panel estimate of V.

Sharp selection bound (Proposition 2). Every selector using only the matched task context satisfies

$$\Delta(g) \;\le\; \underbrace{\mathbb{E}_T \max_a p_{T,a} - \max_a \mu_a}_{\Gamma\text{: best possible task-based gain}} \;\le\; \frac{K-1}{\sqrt{2K}}\sqrt{V}.$$

The first inequality compares a learned selector with an oracle that knows each task's success probabilities. The second bounds that oracle using corrected variation alone. Its coefficient is sharp over the stated model class. In particular, $V = 0$ means that the best fixed setting already matches the task-identity oracle, including when mean success differs across settings. Appendix A states the unbiased estimator, and Appendix B proves the bound.

3.4 Choose a default and assess selection opportunity

First, identify any fixed setting that improves the success–compute trade-off. Then compare the upper confidence limit on selection gain with the improvement that would justify building a selector. A sufficiently small, valid limit can rule out that target; a larger limit leaves it open. Positive dispersion motivates a prediction problem: learning which setting will work best on a held-out task.

We report task-bootstrap intervals and a conservative finite-sample bound. The selection inequality is exact. Its bootstrap confidence limits are approximate, with calibration especially important near zero dispersion. The finite-sample bound supplies the conservative reference: for solver depth, it leaves a ten-point gain open despite a smaller nominal limit. Appendix D gives the calibration and design simulations.

The reported bound covers choosing one fixed setting from task identity. Online decisions and repeat-varying images define richer conditioning contexts. Applying the method at those contexts requires independent continuations matched to the same decision state. Here we evaluate task-level selection. The released tool implements the pair comparisons, unbiased correction, intervals, and decision report from a balanced outcome table.

3.5 Computing the estimator from rollout labels

The disagreement view gives the intuition; a product view gives a compact implementation. For one setting pair, define the observed difference on repeat r as $Z_{t,r} = Y_{t,a,r} - Y_{t,b,r}$. Its expectation is $\delta_{t,ab} = p_{t,a} - p_{t,b}$. Different repeat labels provide independent measurements of that same effect. With R repeats, compute

$$\widehat{H}_{t,ab} = \frac{1}{R(R-1)} \sum_{r \neq s} Z_{t,r} Z_{t,s}, \qquad d_{t,ab} = \frac{1}{R} \sum_{r} Z_{t,r}.$$

Because the factors in each product come from different repeats, $\mathbb{E}[Z_{t,r} Z_{t,s}] = \delta_{t,ab}^2$. Expanding the product recovers the cross-setting disagreement minus within-setting disagreement used above. This is a second-order U-statistic: it averages all distinct-repeat pairs and uses the available rollouts symmetrically.

Centering requires an unbiased estimate of the squared average advantage. Products from different tasks supply that quantity. For a frozen panel of n equally weighted tasks, define

$$U_{ab} = \frac{\sum_{t \neq u} d_{t,ab} d_{u,ab}}{n(n-1)}, \qquad \widehat{V} = \frac{n-1}{n} \binom{K}{2}^{-1} \sum_{a<b} \left(\frac{1}{n} \sum_{t} \widehat{H}_{t,ab} - U_{ab} \right).$$

The factor $(n-1)/n$ makes the estimate target dispersion over the finite panel itself. In expectation, the first term supplies the mean squared effect and the second supplies cross-task products. Together they recover the centered variation exactly; Appendix C gives the short algebraic proof. No task-specific probability model needs to be fitted.

Both pair averages have compact sum-based implementations. Use $\sum_{r \neq s} Z_r Z_s = (\sum_r Z_r)^2 - \sum_r Z_r^2$, and the same identity for the task products. Computation then scales as $O(nK^2R)$ for the full menu. This makes estimation lightweight relative to collecting robot rollouts. Raw negative estimates remain in the output; only the argument of a displayed square-root plug-in ceiling is clipped to zero. An uncertainty interval is computed separately to report estimation precision.

3.6 Why the bound measures selection opportunity

Write each success probability as a default component plus a task-specific residual, $p_{t,a} = \mu_a + q_{t,a}$. The best fixed policy captures the default component. A task-based selector can improve on it only through the residuals. The geometry is simple: a large maximum residual requires the settings to be separated from one another. The average squared pairwise separation of these residuals is exactly V. A centered-vector inequality followed by Cauchy–Schwarz turns that separation into the success-gain ceiling.

This connects an estimable quantity to the objective a router would optimize. It also explains the role of sharpness. Without further structure on task effects, rotating the favored setting across

equally likely tasks attains the coefficient exactly. A two-setting construction alternates success probabilities $(0.8, 0.6)$ and $(0.6, 0.8)$ across two equally likely tasks; it attains a ten-point gain. The methodological advance combines repeat-based effect estimation, unbiased finite-panel centering, and a sharp decision bound in a single compute-selection procedure.

Appendix A extends the same construction to scalar utilities combining success and compute cost.

3.7 How many tasks and repeats reveal an opportunity?

Repeats serve two roles: they make the squared effect estimable, and they increase sensitivity to small task-dependent differences. A simple two-setting model makes the second role precise. Under the null both settings succeed with probability $1/2$. Under the alternative each task draws an independent sign $s \in \{-1, +1\}$, with probabilities $(1 + s\sqrt{\varepsilon})/2$ and $(1 - s\sqrt{\varepsilon})/2$. The average success stays equal across settings while population dispersion is $V = \varepsilon$. Thus the test specifically targets task-dependent opportunity.

With r independent repeats per setting, define

$$q_r(\varepsilon) = \frac{(1+\varepsilon)^{2r} + (1-\varepsilon)^{2r}}{2} - 1.$$

Proposition 3 shows that any test with both error probabilities at most $\eta < 1/2$ requires

$$n \geq \frac{\log(1 + 4(1-2\eta)^2)}{\log(1 + q_r(\varepsilon))}.$$

For fixed r and small ε, the denominator has leading term $r(2r-1)\varepsilon^2$, giving the local necessary scaling $n = \Omega(1/(r^2\varepsilon^2))$. Here ε is a squared probability effect, rather than a success-gain target. In this construction the oracle gain is $\sqrt{\varepsilon}/2$. The rate is a local lower bound for this random-task model; it is not a universal sufficient sample count.

A complementary distribution-free planning rule concerns the uncentered effect S. If each task's corrected statistic lies in $[L_r, U_r]$, set $w_r = U_r - L_r$. Hoeffding's inequality gives a one-sided radius at most e when $n \geq w_r^2 \log(1/\alpha)/(2e^2)$. For three repeats, $[L_r, U_r] = [-1/3, 1]$. This sufficient rule and the local lower bound answer different design questions: conservative precision and the difficulty of detecting small effects. Together they encourage collecting both a broad task panel and repeated attempts on each task.

3.8 From estimated opportunity to a learned selector

The method produces a default-setting comparison and a target for prediction. A favorable fixed setting provides an immediately usable default. When task-dependent opportunity remains, train a selector on separate tasks, freeze it, and evaluate its choices with new rollouts on held-out tasks. Choose the fixed baseline using training data as well, so both methods face the same evaluation set. Report success and compute, including selector overhead. This separates three quantities: a stable effect, the best possible gain from selecting settings, and the gain a learned predictor actually achieves. Our robot evaluation establishes the fixed-setting improvement and supplies the repeat-based estimates that guide this subsequent prediction experiment.

4 Choosing Efficient Compute for Robot Policies

We change two quantities while keeping the checkpoint fixed. Solver depth D is the number of refinement steps used to generate a chunk. Execution horizon E is the number of actions the robot takes from that chunk before observing and replanning. The predicted chunk always contains ten actions; shortening E means discarding its unused tail.

4.1 A stronger fixed policy with fewer policy calls

Success and compute across four settings. The evaluation contains 1,440 episodes: 120 tasks, four settings, and three repeats. The run records confirm one H100, no overlapping runs, and no unexpected runtime errors. The table reports success rates and compute totals over 360 episodes per setting:

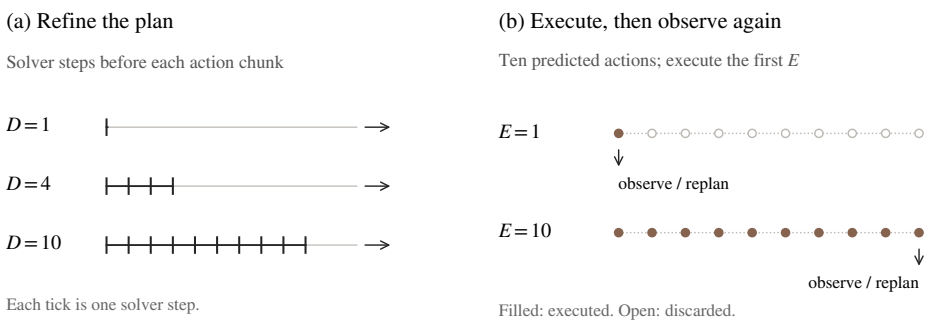


Figure 2: Two compute choices with different control consequences. Left: more solver steps refine the predicted action chunk before execution. Right: a longer execution horizon delays the next observation and policy call. We test $(D, E) = (1, 10), (4, 10), (10, 10), (4, 1)$. The diagrams illustrate the control settings schematically.

Setting	Success (%)	Calls (total)	Inference (s, total)
D=1, E=10	76.39	11,775	476.9
D=4, E=10	78.89	11,308	545.3
D=10, E=10	76.94	11,417	726.3
D=4, E=1	73.06	133,966	6329.3

Executing longer chunks improves the fixed policy. At $D = 4$, executing ten rather than one action before replanning improves success by 5.83 percentage points and reduces policy calls by 91.56%. The one-sided 95% bootstrap lower bounds are 1.39 points and 90.77%, respectively. Thus the measured gain is positive, although the interval does not establish a gain of at least five points. The repeat-corrected squared effect is also positive: 0.0500 (95% interval $[0.0194, 0.0861]$; one-sided lower bound 0.0222). This positive corrected effect identifies a stable change in success probability across execution horizons.

Ten-action execution halves calls relative to five actions. A separately frozen addendum evaluates $D = 4, E = 5$ on the same 120 task identities and three seeds. It achieves 282/360 successes (78.33%), compared with 284/360 (78.89%) for the historical $E = 10$ arm, and uses 63.0 versus 31.4 calls per episode. Ten-action execution saves 50.17% of calls; the nominal one-sided 95% lower bounds are 47.23% for call reduction and -1.94 percentage points for success difference. This passes the prespecified joint screen of at least 15% fewer calls with at most a two-point success loss, narrowly on the success criterion. The two-sided success interval is $[-2.50, 3.61]$ points. This historical-reference result supports a compute-efficient fixed default. Appendix L reports collection provenance, the full three-prefix audit, and uncertainty.

Bounding the opportunity for solver-depth selection. Across $D \in \{1, 4, 10\}$ at $E = 10$, mean corrected pairwise squared effect is -0.0083 (95% interval $[-0.0222, 0.0056]$). Centered heterogeneity is -0.0084, and the nominal one-sided 95% bootstrap task-router ceiling is 4.17 points.

The conservative finite-sample upper bound is 26.72 points, so the 4.17-point bootstrap limit does not establish that a ten-point routing gain is impossible. The registered tests and their interpretation are retained in Appendix H.1.

Compute convention. Policy calls and inference seconds in the arm table are totals over 360 episodes per arm; Figure 3 reports per-episode means.

Four-cell arm menu and frozen design. The 120-task LIBERO-Plus panel balances four suites, camera/language perturbations, and five difficulty levels, and is disjoint from 280 prior identities. On the authenticated public $\pi_{0.5}$ checkpoint we evaluate $(D, E) \in \{(1, 10), (4, 10), (10, 10), (4, 1)\}$, isolating solver depth D and execution prefix E. Three seeds are independent repeats; a prespecified cyclic order runs all $120 \times 4 \times 3 = 1{,}440$ episodes sequentially on one H100. Appendix H gives the complete selection and integrity rules.

Uncertainty and interpretation. Unless labeled finite-sample, intervals are nominal task-cluster bootstrap quantities from 100,000 replicates, with approximate coverage. We report the conservative bound alongside them. The fixed-policy comparison measures an implemented compute change; the routing ceiling measures the opportunity available to a future task-based selector. Each quantity has its own interpretation and uncertainty. Appendix H.1 preserves every registered decision and the infrastructure-monitoring disclosure.

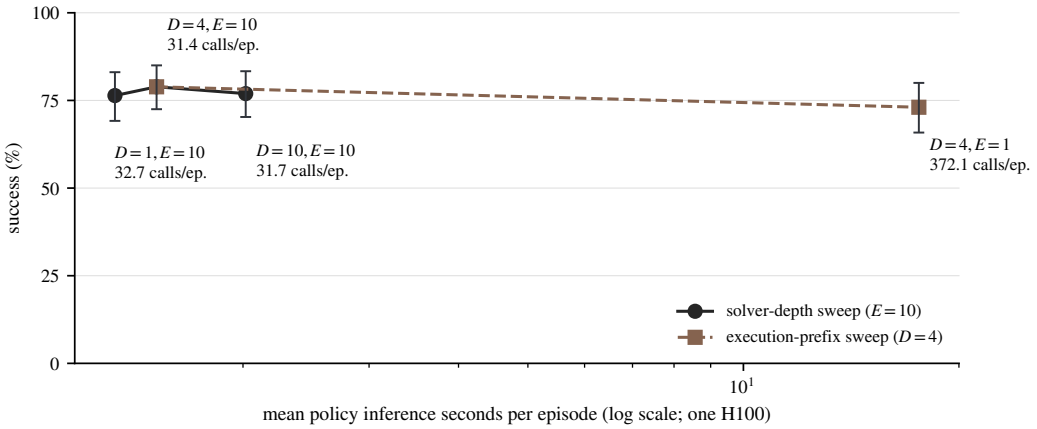


Figure 3: Prospective success–compute frontier for every frozen arm. Labels report policy calls per episode; the horizontal axis is mean policy-inference time per episode on a logarithmic scale. Vertical bars are two-sided 95% percentile intervals from 100,000 task-cluster bootstrap replicates retaining all three seeds; the vertical scale spans 0–100%. Lines show only the prespecified solver-depth contrast at execution prefix 10 and the execution-prefix contrast at solver depth 4.

5 Repeat-Based Effects Across Robot and Agent Tasks

We first reanalyze six already-open panels. Each contains 80 task identities, three repeats per arm, and complete task–setting–repeat tables. This analysis is descriptive: it examines how the estimand separates positive and negative controls across previously collected settings. The panels cover solver depth, execution prefix on three policy/checkpoint families, a failed actuator-fault controller, and the external A^3 controller. Rollouts reaching the terminal horizon with the task incomplete are recorded as failures; every completed row is retained.

Appendix F gives the complete panel table and the cross-arm-versus-repeat-noise decomposition for every targeted contrast.

Estimating solver and controller effects. For depths $\{1, 4, 10\}$, mean corrected pairwise squared difference is -0.0042 (nominal 95% bootstrap interval $[-0.0181, 0.0111]$). The nominal bootstrap task-router ceiling is 7.43 points, even though a naive sample oracle appears to gain 5.0 points. A^3 versus fixed-ten likewise has $H = 0.0083$ ($[-0.0083, 0.0250]$) and a 7.02-point ceiling. Action Echo exactly ties fixed-ten in aggregate; their targeted correction is negative. The positive three-arm menu statistic is driven by the consequential fixed-one versus long-prefix contrast.

Execution prefix survives repeat correction. On the exact public $\pi_{0.5}$ release, the three-prefix mean corrected effect is 0.0250 ($[0.0014, 0.0514]$); the one-versus-ten contrast improves success by 9.58 points. A separately trained direct generator gives 0.1069 ($[0.0597, 0.1597]$) across three prefixes and a 19.58-point one-versus-ten gain. The LoRA transfer panel is directionally consistent but unresolved: $H = 0.0083$ ($[-0.0250, 0.0458]$). These results say that feedback interval can be consequential; predicting the preferred prefix from task identity is the next learning question. Indeed, the retrospective routing ceilings remain wide when corrected heterogeneity is positive.

5.1 Applying the estimator to code and agent outcomes

We apply the same audit to the pinned public MultiPL-E completion artifact (Cassano et al., 2023). The four panels are HumanEval translations in C++, Java, JavaScript, and Rust, each comparing CodeLlama 7B/13B, DeepSeekCoder 1.3B, and StarCoder2 3B at temperature 0.2 and top-p 0.95. Without outcome filtering, we retain the common problems and first 20 stored completions per problem–model cell, yielding 50,880 executions. "Cross-validated same-task selection" selects a model for each task on ten repeats and evaluates it on the other ten, then swaps the halves; it measures repeat-stable selection signal on the evaluated tasks.

We also audit the public repeated-run artifact of Bjarnason et al. (2026): nano-agent and R2E-Gym on SWE-bench Verified, each at source-default and temperature-zero decoding, with three model configurations and ten declared independent runs. Three panels contain all 500 tasks; one uses the 491-task intersection after a run omitted nine submissions, fixed before reading resolution status. This adds 59,730 binary outcomes and an agentic domain with explicit run-level repetition.

Table 1: Public cross-domain portability audit (gains and ceilings in percentage points). Apparent oracle gain—post-hoc selection on a realized draw—exceeds cross-validated same-task selection in all eight panels, yet corrected dispersion is positive. The conservative finite ceilings are wide, leaving selection performance to a subsequent predictive evaluation.

Panel	n	Apparent oracle	CV same-task	Difference	Corrected V	95% ceiling
C++	161	9.1	4.8	4.3	.0710	42.6
Java	158	9.9	6.5	3.4	.0720	42.9
JavaScript	161	13.1	8.1	5.0	.0733	42.8
Rust	156	10.3	5.0	5.3	.0606	41.0
Nano default	500	2.2	-2.8	5.0	.0943	41.3
Nano $T = 0$	500	1.5	-2.0	3.5	.0929	43.2
R2E default	491	14.7	5.1	9.6	.0459	23.8
R2E $T = 0$	500	10.8	1.7	9.1	.0567	26.5

All 36 corrected pairwise effects reject the exchangeable sharp null at their within-panel Bonferroni thresholds. These descriptive reanalyses preserve repeat-stable differences. Appendix J details their sampling provenance and shared-task structure.

6 A Task-Only Selector Across Execution Settings

We fit a ridge predictor on the earlier 80-task depth panel and evaluate its choices on the 120 disjoint prospective task identities. The training panel uses execution prefix five and the evaluation panel uses prefix ten, so this is a descriptive cross-setting transfer test. Word unigram/bigram TF–IDF features of the instruction, suite, and difficulty metadata predict each depth's mean success. Features and weights are fitted on training tasks only; the ridge penalty is fixed at one. The complete procedure was frozen before this fit, after the evaluation outcomes had already informed this paper.

The selector achieves 76.67% success (276/360), compared with 78.89% (284/360) for depth four, selected from training data. The paired difference is -2.22 points, with nominal 95% task-bootstrap interval $[-5.00, 0.28]$. It chooses depths one, four, and ten on 31, 48, and 41 tasks, respectively. On the evaluation panel, the realized same-repeat oracle gain is 6.11 points; the nominal corrected ceiling is 4.17 and the conservative ceiling is 26.72. These quantities describe observed hindsight gain, estimated opportunity, and achieved prediction performance separately. The result leaves useful depth selection open while providing a concrete transfer baseline. Appendix K gives the protocol and claim-audit details.

7 Conclusion

Repeat-to-Select turns repeated outcomes into compute-allocation decisions. Its estimator isolates task-dependent effects, its sharp bound quantifies selection opportunity, and its local lower bound guides the repeat budget. Longer fixed execution improves robot success and reduces policy calls; the five-action comparison supports a compute-efficient ten-action default. Held-out selection measures achieved gain separately. The released outcomes and implementation make the procedure reusable across compute menus.

Reproducibility Statement

Section 3 and Appendices A–H give the estimand, proofs, protocol, complete-row artifact, and independent verifiers; no completed row or failed gate is suppressed. The separately implemented raw-row verifier does not import the primary analyzer and exactly recomputes every confirmatory estimate and gate.

References

Bjarni Haukur Bjarnason, André Silva, and Martin Monperrus. On randomness in agentic evals. *arXiv preprint arXiv:2602.07150*, 2026. URL https://arxiv.org/abs/2602.07150.

Kevin Black, Noah Brown, Danny Driess, Adnan Esmail, Michael 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, James Tanner, Quan Vuong, Anna Walling, Haohuan Wang, and Ury Zhilinsky. π_0: A vision-language-action flow model for general robot control. *arXiv preprint arXiv:2410.24164*, 2024. URL https://arxiv.org/abs/2410.24164.

Kevin Black, Manuel Y. Galliker, and Sergey Levine. Real-time execution of action chunking flow policies. In *Advances in Neural Information Processing Systems*, 2025. URL https://arxiv.org/abs/2506.07339.

Xavier Bouthillier, Pierre Delaunay, Mirko Bronzi, Assya Trofimov, Brennan Nichyporuk, Justin Szeto, Naz Sepah, Edward Raff, Kanika Madan, Vikram Voleti, et al. Accounting for variance in machine learning benchmarks. In *Proceedings of Machine Learning and Systems*, volume 3, 2021. URL https://proceedings.mlsys.org/paper_files/paper/2021/hash/0184b0cd3cfb185989f858a1d9f5c1eb-Abstract.html.

Federico Cassano, John Gouwar, Daniel Nguyen, Sydney Nguyen, Luna Phipps-Costin, Donald Pinckney, Ming-Ho Yee, Yangtian Zi, Carolyn Jane Anderson, Molly Q. Feldman, Arjun Guha, Michael Greenberg, and Abhinav Jangda. MultiPL-E: A scalable and polyglot approach to benchmarking neural code generation. *IEEE Transactions on Software Engineering*, 49(7):3675–3691, 2023. doi: 10.1109/TSE.2023.3267446. URL https://doi.org/10.1109/TSE.2023.3267446.

Feng Chen, Xianghui Wang, Yuxuan Chen, Boying Li, Yefei He, Zeyu Zhang, and Yicheng Wu. Dynamic execution commitment of vision-language-action models. *arXiv preprint arXiv:2605.11567*, 2026. URL https://arxiv.org/abs/2605.11567.

Mark Chen, Jerry Tworek, Heewoo Jun, Qiming Yuan, Henrique Ponde de Oliveira Pinto, Jared Kaplan, Harri Edwards, Yuri Burda, Nicholas Joseph, Greg Brockman, et al. Evaluating large language models trained on code. *arXiv preprint arXiv:2107.03374*, 2021. URL https://arxiv.org/abs/2107.03374.

Jin Cui, Yanbin Hu, Xinyue Long, Linkai Li, Boran Zhao, and Pengju Ren. Look where it matters: Adaptive visual refinement for vision-language-action models. *arXiv preprint arXiv:2608.02197*, 2026. URL https://arxiv.org/abs/2608.02197.

Zhou Fang, Jiaqi Wang, Yi Zhou, and Qiongfeng Shi. ProbeFlow: Training-free adaptive flow matching for vision-language-action models. *arXiv preprint arXiv:2603.17850*, 2026. URL https://arxiv.org/abs/2603.17850.

Xiangdong Feng, Yuxuan Cheng, Chen Shi, Boyao Han, Yuxuan Yan, Yitong Hong, Zhuotao Tian, and Li Jiang. Denoising tells when to replan: Denoising-variance adaptive chunking for flow-based robot policies. *arXiv preprint arXiv:2606.03847*, 2026. URL https://arxiv.org/abs/2606.03847.

Qitian Jason Hu, Jacob Bieker, Xiuyu Li, Nan Jiang, Benjamin Keigwin, Gaurav Ranganath, Kurt Keutzer, and Shriyash Kaustubh Upadhyay. RouterBench: A benchmark for multi-LLM routing system. *arXiv preprint arXiv:2403.12031*, 2024. URL https://arxiv.org/abs/2403.12031.

Audrey Huang, Adam Block, Qinghua Liu, Nan Jiang, Dylan J. Foster, and Akshay Krishnamurthy. Is best-of-N the best of them? coverage, scaling, and optimality in inference-time alignment. *arXiv preprint arXiv:2503.21878*, 2025. URL https://arxiv.org/abs/2503.21878.

Hai Jiang, Yixian Zou, Binbin Liang, Boqian Liu, Fanman Meng, and Shuaicheng Liu. FuturERTC: Real-time robot execution with anticipatory-conditioned action chunking. *arXiv preprint arXiv:2607.24008*, 2026. URL https://arxiv.org/abs/2607.24008.

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.

John P. Lalor, Hao Wu, and Hong Yu. Building an evaluation scale using item response theory. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pp. 648–657. Association for Computational Linguistics, 2016. doi: 10.18653/v1/D16-1062. URL https://aclanthology.org/D16-1062/.

Yuanchang Liang, Xiaobo Wang, Kai Wang, Shuo Wang, Xiaojiang Peng, Haoyu Chen, David Kim Huat Chua, and Prahlad Vadakkepat. Adaptive action chunking at inference-time for vision-language-action models. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2026. URL https://arxiv.org/abs/2604.04161.

Lovish Madaan, Aaditya K. Singh, Rylan Schaeffer, Andrew Poulton, Sanmi Koyejo, Pontus Stenetorp, Sharan Narang, and Dieuwke Hupkes. Quantifying variance in evaluation benchmarks. *arXiv preprint arXiv:2406.10229*, 2024. URL https://arxiv.org/abs/2406.10229.

Andreas Maurer and Massimiliano Pontil. Empirical bernstein bounds and sample variance penalization. In *Proceedings of the 22nd Annual Conference on Learning Theory*, pp. 115–124, 2009. URL https://arxiv.org/abs/0907.3740.

Evan Miller. Adding error bars to evals: A statistical approach to language model evaluations. *arXiv preprint arXiv:2411.00640*, 2024. URL https://arxiv.org/abs/2411.00640.

Junnan Nie, Jiayi Li, Jiachen Zhang, Junyi Lao, Chenghao Liu, Tianle Zhang, Liang Lin, and Songfang Huang. PACE: Phase-aware chunk execution for robot policies with action chunking. *arXiv preprint arXiv:2606.00537*, 2026. URL https://arxiv.org/abs/2606.00537.

Yi Pan, Miao Pan, Qi Lu, Jiaming Huang, Man Zhang, Siteng Huang, Xin Li, Jie Zhang, Yongliang Shen, Xuhong Zhang, and Wenqi Zhang. VLA-corrector: Lightweight detect-and-correct inference for adaptive action horizon. *arXiv preprint arXiv:2607.01804*, 2026. URL https://arxiv.org/abs/2607.01804.

Physical Intelligence. $\pi_{0.5}$: A vision-language-action model with open-world generalization. *arXiv preprint arXiv:2504.16054*, 2025. URL https://arxiv.org/abs/2504.16054.

Yifan Song, Guoyin Wang, Sujian Li, and Bill Yuchen Lin. The good, the bad, and the greedy: Evaluation of LLMs should not ignore non-determinism. *arXiv preprint arXiv:2407.10457*, 2024. URL https://arxiv.org/abs/2407.10457.

Yuewei Sun, Lang Qin, Zechuan Tian, Jingwen Li, Guiqin Wang, Shengzeng Huo, Wenxin Ren, Tao Fang, Xiaochen Zhang, Guanqing Deng, Xiang Wang, Xiaowen Dong, Qinghai Guo, and Yuxin Ma. Fast and accurate: An adaptive VLA inference framework through environment-aware model selection. *arXiv preprint arXiv:2608.06434*, 2026. URL https://arxiv.org/abs/2608.06434.

Gábor J. Székely and Maria L. Rizzo. Energy statistics: A class of statistics based on distances. *Journal of Statistical Planning and Inference*, 143(8):1249–1272, 2013. doi: 10.1016/j.jspi.2013.03.018.

Haoxuan Wang, Gengyu Zhang, Yan Yan, Ramana Rao Kompella, and Gaowen Liu. VLA knows its limits: Adaptive execution horizons for robot policies. In *European Conference on Computer Vision*, 2026a. URL https://arxiv.org/abs/2602.21445.

Rui Wang, Yue Zhang, Jiehong Lin, Kuncheng Luo, Jianan Wang, Zhongrui Wang, and Xiaojuan Qi. When to trust imagination: Adaptive action execution for world action models. *arXiv preprint arXiv:2605.06222*, 2026b. URL https://arxiv.org/abs/2605.06222.

Zihua Wang, Zhitao Lin, Ruibo Li, Yu Zhang, Xu Yang, Siya Mi, and Xiu-Shen Wei. Open-loop planning, closed-loop verification: Speculative verification for VLA. *arXiv preprint arXiv:2604.02965*, 2026c. URL https://arxiv.org/abs/2604.02965.

Weichen Xu, Zhenhua Liu, Lin Luo, Yaobo Liang, Chengtang Yao, Qingyu Mei, Jian Cao, Xixin Cao, Xing Zhang, Jiaolong Yang, and Baining Guo. Continue or replan? Bernoulli-Continuation Policy Learning for adaptive horizon execution. *arXiv preprint arXiv:2608.03483*, 2026. URL https://arxiv.org/abs/2608.03483.

Wenbo Zhang, Hengrui Cai, and Wenyu Chen. Beyond the singular: Revealing the value of multiple generations in benchmark evaluation. *arXiv preprint arXiv:2502.08943*, 2025. URL https://arxiv.org/abs/2502.08943.

AI USE STATEMENT

Generative AI tools contributed to literature search; development and stress-testing of hypotheses, estimands, mathematical claims, and proofs; methodology and experimental design; implementation; data cleaning and analysis; result interpretation; artifact verification; and manuscript drafting and editing. The authors checked the identities, bounds, outputs, and citations and take responsibility for the final content. Figure 1 is a vector schematic with hypothetical outcomes. No AI-generated synthetic observation or outcome is used as experimental evidence.

A FORMAL METHOD AND ESTIMATION RESULTS

We use *arm* as shorthand for a compute setting and *repeat* for another rollout of that setting on the same task and nominal initial state. There are two quantities to keep separate: the size of a setting's effect, and how much that effect varies across tasks. The first tells us whether settings differ; the second limits what adaptive selection could gain.

A.1 SETUP AND UNBIASED EFFECT ESTIMATION

Let T be uniform over a finite panel of $n \geq 2$ tasks. For arm a and repeat r, assume

$$Y_{T,a,r} \mid T = t \sim \text{Bernoulli}(p_{t,a}).$$

Distinct tasks are independent sampling units. Within a task, outcomes with different repeat labels are conditionally independent across arms, and the repeat distribution is stable within each task–arm cell. Same-label cross-arm outcomes may be coupled and are never used in the identifying contrast. The probabilities may otherwise vary arbitrarily. In particular, we do not assume equal arm means, arm exchangeability, monotone compute effects, independent same-seed arms, or a one-dimensional task difficulty.

Proposition 1 (No unbiased one-shot estimator)**.** *For an arbitrary fixed finite panel, one independent binary observation per task and arm admits no estimator that is unbiased for $\mathbb{E}_T[(p_{T,a} - p_{T,b})^2]$ at every collection of cell probabilities, even if the two arms' outcomes are conditionally independent given task.*

Proof. For any statistic of the one-shot binary panel, its expectation is a multilinear polynomial in the cell probabilities: each $p_{t,a}$ and $p_{t,b}$ has degree at most one because that cell contributes only one Bernoulli draw. The target equals $n^{-1}\sum_t (p_{t,a}^2 + p_{t,b}^2 - 2p_{t,a}p_{t,b})$ and has a nonzero quadratic term in every cell probability. No multilinear polynomial can equal this target on $[0,1]^{2n}$. Thus no one-shot statistic is design-unbiased over the unrestricted finite-panel model. Repeats supply the within-arm products that the missing quadratic terms require. □

This is an unbiased-estimation impossibility, not statistical non-identifiability: distinct cell probabilities define distinct observable laws. It does not exclude biased estimators, structured pooling, Bayesian inference, or useful one-shot decisions under additional assumptions.

Lemma 1 (Repeat-difference identity)**.** *For $a \neq b$, compare different repeats $r \neq s$ and define*

$$C_{t,ab} = \mathbb{P}(Y_{t,a,r} \neq Y_{t,b,s}),$$
$$R_{t,a} = \mathbb{P}(Y_{t,a,r} \neq Y_{t,a,s}).$$

Then

$$H_{t,ab} := C_{t,ab} - \frac{R_{t,a} + R_{t,b}}{2} = (p_{t,a} - p_{t,b})^2. \tag{1}$$

Proof. Conditional independence gives $C_{t,ab} = p_{t,a} + p_{t,b} - 2p_{t,a}p_{t,b}$ and $R_{t,a} = 2p_{t,a}(1 - p_{t,a})$. Substitution leaves $p_{t,a}^2 + p_{t,b}^2 - 2p_{t,a}p_{t,b}$. □

Corollary 1 (Best-of-K is not one-arm routing evidence)**.** *If K conditionally independent arms have the same success probability p_t on instance t, their true task-routing dispersion is zero, while a hindsight maximum succeeds with probability $1 - (1 - p_t)^K$. Thus the latter quantity is valid for a system that executes and verifies all K arms, but its gain over p_t cannot justify selecting one arm before observing outcomes.*

Corollary 2 (Scalar-return extension)**.** *Let $X_{t,a,r}$ be any square-integrable scalar return with mean $m_{t,a}$, with independent stable repeats. Define $C_{t,ab} = \mathbb{E}[(X_{t,a,r} - X_{t,b,s})^2]$ and $R_{t,a} = \mathbb{E}[(X_{t,a,r} - X_{t,a,s})^2]$ for $r \neq s$. Then*

$$C_{t,ab} - \frac{R_{t,a} + R_{t,b}}{2} = (m_{t,a} - m_{t,b})^2.$$

Proof. The cross-arm squared distance is $\mathrm{Var}(X_{t,a}) + \mathrm{Var}(X_{t,b}) + (m_{t,a} - m_{t,b})^2$, while each within-arm squared distance is twice that arm's variance. Subtraction cancels both variances. □

For binary returns, squared distance equals disagreement and Corollary 2 reduces to Lemma 1. For normalized scalar utilities, replacing p by m in the results below bounds expected-utility routing gain.

With three repeats, we estimate $R_{t,a}$ from all three unordered within-arm pairs and $C_{t,ab}$ from the six cross-arm pairs whose repeat labels differ. Equal seed labels are excluded because paired simulator randomness can couple arms even when different seeds are independent.

The matching unit can be an online state. Nothing in Lemma 1 requires t to denote a whole task. If a replayable simulator snapshot supplies independent downstream continuations from the same decision context x, replacing t by x identifies state-conditional arm effects and bounds routers that observe that context. Our experiments use task–nominal-initial-state identity, averaging over repeat-varying sensory inputs. Thus the reported ceilings cover task-only routing. Extending them to routers using fresh initial images requires repeats matched at that image.

A.2 How much could task-based selection help?

For K arms, write $\mu_a = \mathbb{E}_T p_{T,a}$ and $q_{t,a} = p_{t,a} - \mu_a$. Define average centered pairwise dispersion

$$V = \binom{K}{2}^{-1} \sum_{a<b} \mathbb{E}_T \left[((p_{T,a} - p_{T,b}) - (\mu_a - \mu_b))^2 \right]. \tag{2}$$

Proposition 2 (Sharp pre-execution router ceiling)**.** *The task-identity oracle's gain over the best fixed arm,*

$$\Gamma = \mathbb{E}_T \max_a p_{T,a} - \max_a \mu_a,$$

satisfies

$$\boxed{\Gamma \leq \frac{K-1}{\sqrt{2K}} \sqrt{V}.} \tag{3}$$

The coefficient is jointly sharp.

Proof sketch. For any real vector x, centering and Cauchy–Schwarz give

$$\max_a x_a - \bar{x} \le \frac{K-1}{\sqrt{2K}} \sqrt{\binom{K}{2}^{-1} \sum_{a<b} (x_a - x_b)^2}.$$

Apply this taskwise to q_t, then Jensen's inequality. Equality is attained by rotating $((K-1)c, -c, \ldots, -c)$ uniformly over arms across tasks, with equal arm means; choosing mean $1/2$ and $0 < c \le 1/[2(K-1)]$ keeps all probabilities feasible. Appendix B gives the complete argument. □

Any pre-execution selector whose inputs are deterministic functions of the matched context t chooses one arm per t and is therefore dominated by this oracle. This includes selectors based on the instruction, its embedding, the fixed initial observation, task metadata, or frozen features thereof. Here "fixed" is essential: recorded initial observations vary across repeats. A fresh stochastic probe or within-episode observation enlarges the information set and is not covered without rematching at that decision context.

Cost-aware consequence. Let arm a have a task-independent deployment cost c_a and define utility $u_{t,a} = p_{t,a} - \lambda c_a$ for any $\lambda \ge 0$. Centering cancels the constant difference $c_a - c_b$, so Eq. 2 and the same sharp ceiling hold simultaneously for every λ. Thus a small success-effect dispersion also limits task-dependent improvement over the best fixed arm on every such success–cost scalarization. When cost itself varies by task or rollout, Corollary 2 instead applies directly to a prespecified repeated scalar utility combining success and realized cost.

The centering in V is crucial. A large constant gap between arms makes arm choice important but does not justify task-dependent routing: the best fixed arm already captures it. The bound measures only how the arm differences vary over tasks.

A.3 Unbiased finite-panel estimator

Let $\widehat{H}_{t,ab}$ be the sample version of Eq. 1, and let $d_{t,ab} = \bar{Y}_{t,a} - \bar{Y}_{t,b}$. The cross-task U-statistic

$$U_{ab} = \frac{1}{n(n-1)} \sum_{t \ne u} d_{t,ab} d_{u,ab}$$

is unbiased for the average cross-task product of true effects. Therefore

$$\widehat{V}_{ab} = \frac{n-1}{n} \left(\frac{1}{n} \sum_t \widehat{H}_{t,ab} - U_{ab} \right) \tag{4}$$

is unbiased, over rollout repetition, for the finite-panel variance of $p_{t,a} - p_{t,b}$. We average over arm pairs, cluster-bootstrap tasks, retain negative estimates, and truncate only the argument of the final square root.

A.4 Finite-sample robustness and design resolution

We add seed-block arm-swap tests and a conservative empirical-Bernstein ceiling as secondary, finite-sample checks; neither changes the registered bootstrap gates. Appendix D states their additional exchangeability condition, bound, design rule, and coverage–power simulation.

B Complete Proof of the Routing Ceiling

For $x \in \mathbb{R}^K$, let $\bar{x} = K^{-1} \sum_a x_a$ and let j index a maximum. Since the centered coordinates sum to zero,

$$x_j - \bar{x} = -\sum_{a \ne j} (x_a - \bar{x}).$$

Cauchy–Schwarz gives $(x_j - \bar{x})^2 \le (K-1) \sum_{a \ne j} (x_a - \bar{x})^2$ and hence

$$(x_j - \bar{x})^2 \le \frac{K-1}{K} \sum_a (x_a - \bar{x})^2.$$

The pairwise identity

$$\sum_a (x_a - \bar{x})^2 = \frac{1}{K}\sum_{a<b}(x_a - x_b)^2 = \frac{K-1}{2}\binom{K}{2}^{-1}\sum_{a<b}(x_a - x_b)^2$$

proves the deterministic inequality used in Proposition 2. With $\mu_* = \max_a \mu_a$,

$$\max_a p_{t,a} = \max_a(\mu_a + q_{t,a}) \le \mu_* + \max_a q_{t,a}.$$

Averaging over tasks, subtracting μ_*, applying the deterministic inequality to q_t, and then Jensen's inequality yields Eq. 3. If tasks rotate $((K-1)c, -c, \ldots, -c)$ uniformly across coordinates and all arm means are equal, every inequality is an equality. Taking common mean $1/2$ and $0 < c \le 1/[2(K-1)]$ places the construction in $[0,1]^K$, so the coefficient is sharp even for success probabilities.

C Why the Finite-Panel Estimator Is Unbiased

Let $\delta_{t,ab} = p_{t,a} - p_{t,b}$. Lemma 1 gives $\mathbb{E}\widehat{H}_{t,ab} = \delta^2_{t,ab}$. Independent tasks give $\mathbb{E}[d_{t,ab}d_{u,ab}] = \delta_{t,ab}\delta_{u,ab}$ for $t \neq u$. Therefore

$$\mathbb{E}\left[\frac{1}{n}\sum_t \widehat{H}_{t,ab} - U_{ab}\right] = \frac{1}{n}\sum_t \delta^2_{t,ab} - \frac{1}{n(n-1)}\sum_{t\neq u}\delta_{t,ab}\delta_{u,ab}.$$

Multiplication by $(n-1)/n$ equals the variance under the uniform empirical distribution over the n frozen tasks. Averaging over pairs yields $\widehat{V}$.

D Finite-Sample Robustness and Design Resolution

The registered analysis uses a task-cluster bootstrap. We add two secondary checks that do not alter its frozen gates. First, for a pair of binary arms under the sharp null $p_{t,a} = p_{t,b}$, swap arm labels independently inside each matched task–seed block. This preserves same-seed dependence and gives an exact randomization distribution when all swaps are enumerated, or a finite-sample Monte Carlo test with the standard plus-one correction. It requires pairwise arm-label exchangeability under the sharp null; the primary repeat-difference identity does not.

Second, let Z_t average $\widehat{H}_{t,ab}$ over all arm pairs and let $S = n^{-1}\sum_t \mathbb{E}Z_t$ be the finite-panel uncentered mean squared effect. For r repeats, exhaustive binary enumeration supplies deterministic bounds $Z_t \in [L_r, U_r]$ (for $r = 3$, $[-1/3, 1]$). With independent task clusters, the empirical-Bernstein inequality (Theorem 11 of Maurer & Pontil, 2009, allowing nonidentical distributions) gives, with probability at least $1-\alpha$,

$$S \le \bar{Z} + \sqrt{\frac{2s_Z^2\log(2/\alpha)}{n}} + \frac{7(U_r - L_r)\log(2/\alpha)}{3(n-1)} =: U_S. \tag{5}$$

Because centered dispersion satisfies $V \le S$, Proposition 2 yields $\Gamma \le \frac{K-1}{\sqrt{2K}}\sqrt{\max(0, U_S)}$. This finite-sample certificate is more conservative than the centered bootstrap limit but remains valid at the zero-variance boundary.

Prospective finite bound and post-hoc calibration check. An initial-context audit of all 1,440 hash-verified rows finds matching initial-state indices but differing recorded features on 77 of 120 tasks: base-image summaries on 74, wrist-image summaries on 74, and robot-state vectors on 3 (overlapping counts). Full-image equality is not established. Consequently these reported ceilings average over observation variation and do not bound routers exploiting fresh initial observations.

Recomputing this bound from the hash-verified prospective rows gives depth and prefix ceilings of 26.72 and 22.02 points, respectively. Neither excludes a 10-point gain. These checks do not change the frozen bootstrap decisions; they limit their interpretation. The exact identity and oracle inequality must not be conflated with calibration of an estimated confidence bound.

A separate post-hoc synthetic check uses 120 fixed tasks, three arms and three independent Bernoulli repeats. Arm probabilities rotate $(b + 2a, b - a, b - a)$ evenly over tasks, with $b \in \{.5, .75\}$ and $a \in \{0, .025, .075, .15\}$; the invalid $(.75, .15)$ cell is excluded without clipping. In 500 panels per valid cell, using 1,000 bootstrap draws per panel, coverage of the nominal 95% upper limit for nonzero true dispersion ranges from 91.8% to 94.0% (Monte Carlo standard errors 1.06–1.23 points). Zero-dispersion coverage is automatic after truncation and is uninformative. This exploratory check uses fewer bootstrap draws than the frozen 100,000; it is not a calibration proof or a replacement gate. It motivates reporting the 4.17-point value as nominal, not a finite-sample certificate.

An independent-seed follow-up holds $a = .025$ and tests both base rates on 500 new panels each. Nested 1,000/10,000 bootstrap draws give coverage 94.4/94.2% at $b = .5$ and 93.4/93.2% at $b = .75$; only one coverage decision changes per cell. Standard errors for the 10,000-draw estimates are 1.05 and 1.13 points. The draw-count sensitivity is small in these cells, but Monte Carlo uncertainty precludes a precise or universal calibration claim.

Corollary 3 (A repeat-first design rule)**.** *Writing $w_r = U_r - L_r$, Hoeffding's inequality implies that $n \geq w_r^2 \log(1/\alpha)/(2\varepsilon^2)$ matched units suffice to make the one-sided, distribution-free radius for S at most ε. Repeats matter twice: two enable unrestricted unbiased squared-effect estimation, and increasing r narrows w_r and the realized variance. This is a sufficient planning bound; the following result gives a separate local necessity statement under an explicit random-task model.*

Proposition 3 (A local repeat-complexity lower bound)**.** *Consider the two-arm i.i.d. random-task subclass with r independent binary repeats per arm. Under the null, $p_a = p_b = 1/2$. Under the alternative, each task independently draws $S \in \{-1, +1\}$ uniformly and sets $p_a = (1 + S\sqrt{\varepsilon})/2$ and $p_b = (1 - S\sqrt{\varepsilon})/2$, for $0 < \varepsilon < 1$. The two marginal arm means remain equal and the population centered dispersion is $V = \varepsilon$. Define*

$$q_r(\varepsilon) = \frac{(1+\varepsilon)^{2r} + (1-\varepsilon)^{2r}}{2} - 1.$$

Any test with type-I and type-II error both at most $\eta < 1/2$ requires

$$n \geq \frac{\log\big(1 + 4(1-2\eta)^2\big)}{\log(1 + q_r(\varepsilon))}. \tag{6}$$

For fixed r and $\varepsilon \downarrow 0$, $q_r(\varepsilon) = \binom{2r}{2}\varepsilon^2 + O(\varepsilon^4)$, so locally $n = \Omega((r^2\varepsilon^2)^{-1})$.

Proof. Encode each null Bernoulli outcome as a Rademacher variable. Conditional on task sign s, the likelihood factor is $1 + s\sqrt{\varepsilon}x$ for an arm-a observation and $1 - s\sqrt{\varepsilon}y$ for arm b. If L_s is the product over the $2r$ observations, then under the null $\mathbb{E}_0[L_s L_t] = (1 + st\varepsilon)^{2r}$. The alternative likelihood ratio is $L = (L_+ + L_-)/2$, hence $\chi^2(P_1 \| P_0) = \mathbb{E}_0[L^2] - 1 = q_r(\varepsilon)$. Independence across tasks gives $\chi^2(P_1^{\otimes n} \| P_0^{\otimes n}) = (1 + q_r)^n - 1$. Le Cam's inequality and $\mathrm{TV}(P_1^{\otimes n}, P_0^{\otimes n}) \leq \frac{1}{2}\sqrt{\chi^2(P_1^{\otimes n} \| P_0^{\otimes n})}$ imply Eq. (6). Averaging the two signs gives equal arm means and squared difference ε. The expansion retains the first even binomial term. □

This proposition is a minimax design statement for an i.i.d. random-task subclass, not an assumption used by the paper's finite-panel identity or prospective certificate. It supplies a necessary local rate while preserving the main analysis's outcome- and distribution-free target.

We simulate balanced two-arm panels with base success 0.5, sign-changing instance effects, $n \in \{30, 60, 120, 240\}$, $r \in \{2, 3, 5\}$, and centered dispersion $V \in \{0, 0.01, 0.04\}$. Each cell uses 200 independently generated panels; randomization tests use 499 arm-swap draws. The finite ceiling covered the sharp true ceiling in every simulated cell (conservative, not evidence of exact 100% coverage), while null rejection stayed near or below 5%. At $V = 0.04$, power at $n = 120$ rises from 12.0% with two repeats to 75.5% with five; at five repeats it rises from 25.0% at $n = 30$ to 97.5% at $n = 240$.

E Master Inventory of Repeated Panels

Table 2 lists every panel used for a reported result. The four code-language rows share translated HumanEval identities and require joint interpretation; the four agent rows likewise share SWE-bench

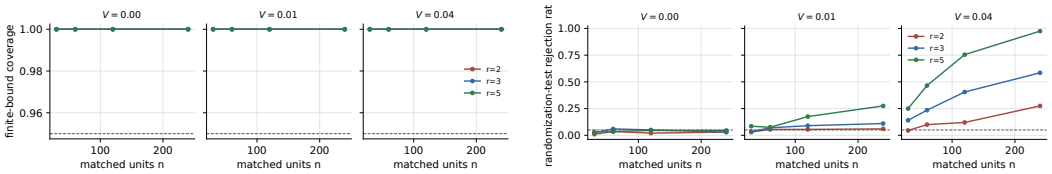


Figure 4: Outcome-independent design simulation. Left: coverage of the finite-sample router ceiling; the dashed line is 95%. Right: rejection rate of the seed-block randomization test; at $V = 0$ this is type-I error and for $V > 0$ it is power.

tasks and model configurations. "Public batch" denotes the missing per-completion seed provenance detailed in Appendix J.

Table 2: All auditable multi-repeat panels. n, K, and r are matched units, arms, and repeats per cell.

Panel	Domain	n	K	r	Evidence role / repeat provenance
Depth	VLA	80	3	3	Descriptive; explicit seeds
Exact prefix	VLA	80	3	3	Descriptive; explicit seeds
LoRA prefix	VLA	80	2	3	Descriptive; explicit seeds
Direct prefix	VLA	80	3	3	Descriptive; explicit seeds
Fault menu	VLA	80	3	3	Descriptive; explicit seeds
A^3 vs fixed	VLA	80	2	3	Descriptive; explicit seeds
MultiPL-E C++	Code	161	4	20	Portability; public batch
MultiPL-E Java	Code	158	4	20	Portability; public batch
MultiPL-E JavaScript	Code	161	4	20	Portability; public batch
MultiPL-E Rust	Code	156	4	20	Portability; public batch
Nano default	Agent	500	3	10	Portability; declared runs
Nano $T = 0$	Agent	500	3	10	Portability; declared runs
R2E default	Agent	491	3	10	Portability; declared runs
R2E $T = 0$	Agent	500	3	10	Portability; declared runs
Two-axis $\pi_{0.5}$	VLA	120	4	3	Confirmatory; sealed explicit seeds

F Full Retrospective Panel Summary

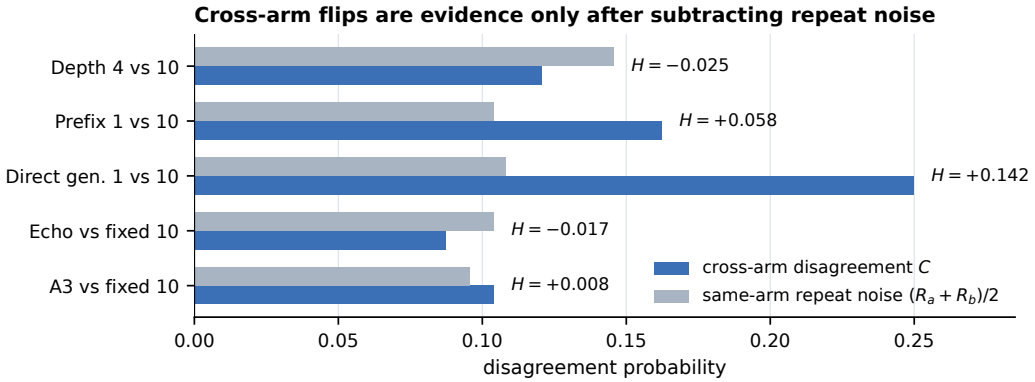


Figure 5: Targeted contrasts from already-open three-repeat panels. Raw cross-arm disagreement C is not an effect: subtracting the two arms' repeat noise yields $H = (p_a - p_b)^2$. Depth 4 vs. 10 and Action Echo vs. fixed 10 have less cross-arm disagreement than repeat noise. Prefix interventions on the exact release and a direct generator survive the correction.

External-controller provenance. The A^3 row executes the authors' 10-step controller at commit `96a85c698f3848f6e684e36a81caf6e275db2ecc` after checkpoint-export parity checks against the public $\pi_{0.5}$ release. Its batch-eight draft/verification calls are not equivalent to fixed-prefix policy calls; we therefore use matched binary success in the repeat decomposition but do not present its controller decisions as fixed-arm compute savings.

Table 3: All six previously inspected panels. $\overline{H}$ averages corrected squared effects over all arm pairs. Ceiling is the one-sided 95% upper bound on task-oracle gain over the best fixed arm.

Panel	Arms (success rates, %)	Naive oracle (pp)	$\overline{H}$	95% interval	Ceiling (pp)
Depth	1: 74.17; 4: 76.25; 10: 75.42	5.00	$-.0042$	$[-.0181, .0111]$	7.43
Exact prefix	1: 71.67; 5: 72.50; 10: 81.25	2.92	.0250	$[.0014, .0514]$	16.18
LoRA prefix	1: 71.67; 10: 75.42	4.58	.0083	$[-.0250, .0458]$	9.54
Direct prefix	1: 43.33; 5: 49.17; 10: 62.92	3.33	.1069	$[.0597, .1597]$	28.13
Fault menu	echo: 75.42; short: 65.42; long: 75.42	5.00	.0431	$[.0097, .0819]$	20.10
A^3 vs fixed	A^3: 75.83; long: 81.25	1.67	.0083	$[-.0083, .0250]$	7.02

G Prospective Descriptive Strata

Table 4 reports every group frozen before outcomes: the overall panel, all four suites, both perturbation roles, and all five difficulty levels. These are point estimates only; no subgroup interval, test, gate, or selection claim is made. The confirmatory pooled analysis remains authoritative.

Table 4: All prespecified prospective strata without suppression. Success columns are percentages. Depth bound is the plug-in point ceiling for the three long-prefix solver arms; prefix gain is $E = 10$ minus $E = 1$ at $D = 4$; call reduction compares the same prefix pair.

Group	n	$D1, E10$	$D4, E10$	$D10, E10$	$D4, E1$	Depth bound	Prefix gain	Calls red.
All tasks	120	76.4	78.9	76.9	73.1	0.0	+5.8	91.6
Suite: spatial	30	78.9	81.1	78.9	82.2	0.0	-1.1	90.1
Suite: object	30	81.1	88.9	85.6	82.2	3.2	+6.7	92.5
Suite: goal	30	77.8	75.6	73.3	65.6	0.0	+10.0	92.2
Suite: LIBERO-10	30	67.8	70.0	70.0	62.2	3.2	+7.8	91.2
Role: camera	60	67.2	71.1	67.8	68.3	0.0	+2.8	91.0
Role: language	60	85.6	86.7	86.1	77.8	0.0	+8.9	92.2
Difficulty 1	24	87.5	90.3	88.9	87.5	0.0	+2.8	92.2
Difficulty 2	24	81.9	79.2	77.8	81.9	0.0	-2.8	89.3
Difficulty 3	24	83.3	81.9	83.3	80.6	0.0	+1.4	91.2
Difficulty 4	24	75.0	80.6	76.4	62.5	1.3	+18.1	92.9
Difficulty 5	24	54.2	62.5	58.3	52.8	0.0	+9.7	91.5

H Prospective Protocol and Integrity

The panel file is content-addressed with SHA-256 `1bf50f4878b681fadf57bbce6d34347ec73e2e873e0820b7434387a2ad588d47`. The frozen protocol is `5c9f0b310a2dc7051e8031d819808c2709ab43743bc85013339f782d7c1b5384`. The checkpoint is the official OpenPI `pi05_libero` tree at release commit `35106c9bea0f84584e1a9b3a9ee1ae7aba070532`. A create-once launcher rejects dirty source trees, wrong checkpoints, altered inputs, non-H100 devices, more than one visible GPU, busy reserved hardware, duplicate arm roots, and any arm outside the four frozen cells. A global lock prevents concurrent policy servers. A blind verifier checks exact row pairing, task assets, run intervals, the single GPU UUID, and zero unexpected errors before allowing aggregate analysis.

The prespecified arm order is:

1. seed 211: $(1, 10), (4, 10), (10, 10), (4, 1)$;
2. seed 223: $(4, 10), (10, 10), (4, 1), (1, 10)$;
3. seed 227: $(10, 10), (4, 1), (1, 10), (4, 10)$.

H.1 Registered decisions and interpretation

Frozen decisions and interval convention. Unless labeled finite-sample, reported intervals and limits are nominal task-cluster bootstrap quantities, not exact coverage guarantees. The frozen depth decision is reported here as a *nominal screen*: it passes when the bootstrap one-sided 95% ceiling is below 10 points. Finite-sample certification is reported separately; its depth ceiling exceeds 10 points, leaving that target unresolved. This terminology preserves the original decision rule and

distinguishes its evidential role. The fixed-prefix gates require a positive lower bound for corrected H, a success-gain point estimate of at least five points with a lower bound above zero, and at least 80% fewer policy calls in both the point estimate and lower bound. They do not require a five-point success lower bound. The task-cluster bootstrap uses 100,000 replicates; seed-block tests and the finite bound are secondary checks. A disagreement is insufficient resolution at the registered threshold; the original gate remains in force.

Registered nominal criterion	Decision
Depth bootstrap screen < 10 pp	nominal pass
Prefix corrected H lower bound > 0	pass
Prefix success gain ≥ 5 pp	pass
Prefix success lower bound > 0	pass
Call reduction $\geq 80\%$	pass
Call-reduction lower bound $\geq 80\%$	pass

The registered bootstrap criteria pass as nominal screens. The separate finite-sample depth certificate remains inconclusive (Appendix D); no new criterion was registered retrospectively. One first-row success flag was incidentally disclosed during infrastructure monitoring after all choices were frozen; no execution or analysis choice was changed, and all 1,440 rows remained mandatory.

I Assumption Audit

Table 5: What is required, checked, and outside scope.

Item	Evidence	Boundary
Independent repeats	Fresh policy RNG seeds; different seed labels for cross-arm pairs	Does not cover hidden shared randomness outside the sealed runner
Stable task–arm law	Exact checkpoint, task, init state, horizon, and source hashes	Distribution shift is not covered
One GPU	UUID binding, one visible JAX H100, non-overlapping intervals	Wall clock contention outside the reserved GPU may affect latency
Task-only routing	Sharp ceiling over the frozen arm menu	Phase controllers with within-episode evidence can exceed this information set
Binary success	Unchanged LIBERO done signal	Path safety and manipulation quality are unmeasured

J Additional Scope and Experimental Details

The identity requires independent, stable repeats. Shared simulator forks, common random numbers, or seed-dependent policy changes add covariance terms. Two repeats identify the effect but can leave wide intervals; negative estimates must remain visible. The ceiling covers only the frozen arm menu and task panel. Binary success omits path quality and safety; scalar utilities are covered, but vector objectives need a deployment utility. Bootstrap coverage is asymptotic near the zero-variance boundary. The finite bound is valid but conservative, while its randomization test adds sharp-null exchangeability. The bootstrap resamples 120 task clusters without fixing the 40 stratum counts and is not randomization-exact for panel selection. Retrospective panels were inspected before defining this estimand and are descriptive. One prospective first-row success flag was visible during infrastructure monitoring after the design and analysis freeze. The evidence package records this deviation; all 1,440 rows remained mandatory. We claim pre-frozen, non-adaptive analysis rather than continuous outcome blinding. MultiPL-E completions lack per-draw seeds and may share batches; language panels share translated HumanEval tasks. Agent runs are source-declared independent but lack an authenticated seed chain, and their panels share SWE-bench tasks and models. Both lookups observe earlier outcomes for the same task, not unseen-task routing. Their oracle–selection gap includes finite-training-repeat selection error, not just noise inflation; it does not rule out better routers. Positive dispersion does not establish predictable unseen-task winners. One panel excludes nine missing submissions before reading resolution status.

K Post-Hoc Selector Transfer and Published-Claim Audit

Frozen fit and evaluation. Training uses 80 earlier task identities at $E = 5$, depths $\{1, 4, 10\}$, and seeds $\{47, 53, 59\}$; evaluation uses 120 disjoint identities at $E = 10$ and seeds $\{211, 223, 227\}$. Related instruction/layout families may remain across panels. The lexical features are training-fitted TF–IDF word unigrams and bigrams, concatenated with one-hot suite and difficulty labels. A multi-output ridge regression with penalty one and LSQR tolerance 10^{-10} predicts the three training mean-success labels. The largest predicted value selects a depth; ties favor the smaller depth. The fixed baseline is likewise chosen on training data. No hyperparameter search is performed. The protocol and code are hashed before this fit, and predictions are saved before computing the evaluation summaries. Both panels were previously inspected, so this is post-hoc transfer evidence, not a newly blinded test or a matched-execution-setting comparison.

The training means for depths one, four, and ten are 74.17%, 76.25%, and 75.42%; evaluation means are 76.39%, 78.89%, and 76.94%. The selector scores 76.67%. We resample 120 task clusters, retaining each task's three repeats, for 100,000 draws with seed 20260917. The interval is conditional on the fitted predictor. Fresh prediction refitting and raw-row summation reproduce 276 selector successes and 284 fixed-depth-four successes. The experiment uses CPU computation and zero additional robot rollouts. It supplies a simple lexical baseline; pretrained semantic features remain a separate modeling choice.

Published claims require matched interpretations. Our source audit distinguishes online-controller performance from task-identity selection. AAC reports 50 LIBERO and 100 RoboCasa rollouts per task (Liang et al., 2026). Its Table 1 gives average gains of 0.9 and 2.3 points over the default horizon; comparison with the strongest listed fixed horizon reduces these to 0.3 and 0.8 points. This is a baseline-sensitivity calculation, not a repeat-corrected significance test. AutoHorizon reports 25 independent LIBERO rollouts per task and three repeated experiments (Wang et al., 2026a). Its attention-conditioned online decisions use a richer information set than our task-only selector. A^3 v4 uses a quantized verifier, making exact version matching essential (Chen et al., 2026). Aggregate tables alone do not establish the matched outcome structure needed for our correction. These sources therefore motivate careful baseline and information-set comparisons; we make no claim that their published controller gains disappear under repeat correction.

L Five-Action Execution Addendum

Design and provenance. After inspecting the original four-arm results, we froze a separate analysis plan for $(D, E) = (4, 5)$ and collected 360 episodes on September 17, 2026. The original $E = 1$ and $E = 10$ reference runs were collected in August 2026. All 120 task identities, initial-state indices, task-definition hashes, initial-state file hashes, and selected state-payload hashes match the original panel. Seeds 211, 223, and 227 run sequentially on one H100. The authenticated checkpoint, OpenPI revision, action caps, ten warmup actions, and rollout loop are preserved. The simulator is restored from the recorded revision and compatibility patch; the downloaded asset archive matches the publisher's checksum. The new run uses a different physical H100, so policy-call counts are the compute comparison and elapsed inference time is descriptive. Collection date is confounded with the new setting; matched assets and seeds support a historical-reference comparison, not a concurrently randomized experiment. The plan was frozen locally before the new outcomes, not externally registered. Original results remain unchanged.

Prefix at $D = 4$	Successes	Success (%)	Calls	Inference (s)
$E = 1$ (historical)	263/360	73.06	133,966	6329.3
$E = 5$ (addendum)	282/360	78.33	22,692	1156.9
$E = 10$ (historical)	284/360	78.89	11,308	545.3

Prespecified practical comparison. The joint screen requires the one-sided nominal 95% task-bootstrap lower bound for $E = 10$ minus $E = 5$ success to be at least -2 percentage points, and the corresponding lower bound on call reduction to be at least 15%. We resample 120 task clusters, preserving all three repeat labels, for 100,000 draws with seed 20260918. Call reduction is one

minus the ratio of aggregate calls, rather than the mean of task-wise ratios. The success difference is $+0.56$ points (two-sided 95% interval $[-2.50, 3.61]$); its one-sided lower bound is -1.94 points. Call reduction is 50.17%, with lower bound 47.23%. Both criteria pass. The success margin is only 0.06 points inside the prespecified threshold, so the finding supports that nominal screen rather than success equivalence or superiority. All episodes are retained, including 23 unsuccessful terminal-horizon timeouts (9, 7, and 7 by seed); no unexpected runtime errors are recorded.

Three-prefix opportunity. For the menu $E \in \{1, 5, 10\}$, the repeat-corrected squared effect is $H = 0.02685$ (nominal 95% interval $[0.00648, 0.05000]$), and centered heterogeneity is $V = 0.02510$ ($[0.00576, 0.04525]$). Signed estimates are retained before forming ceilings. The nominal one-sided centered routing ceiling is 16.68 points; the conservative empirical-Bernstein ceiling is 31.98 points, using $(K-1)/\sqrt{2K}$ with $K = 3$. These menu-level statistics include the large $E = 1$ contrast and do not isolate routing opportunity between five and ten actions. They describe task-identity opportunity under the stated repeat assumptions; first-frame matching and a predictive selector remain separate questions.

For comparison, the original two-prefix menu has centered heterogeneity $V = 0.0469$ (nominal 95% interval $[0.0169, 0.0790]$), a point-estimate ceiling of 10.83 points, and a nominal upper ceiling of 13.55 points. Changing the menu changes the estimand, so these ceilings are not a ranking of selector quality. The E5 derived panel and a standalone NumPy reproducer are included with the manuscript source.