ADAPTING LEARNED ROBOT SIMULATORS: POLICY VISITATION BEATS ERROR MAGNITUDE

Anonymous authors
Paper under double-blind review

ABSTRACT

Policies trained inside learned robot simulators exploit the simulators' errors: on the public NeRD model of the Ant robot, jump policies score $7.9\times$ higher in the model than in the physics engine it imitates. The prevailing remedy is to adapt the simulator with data collected where the model errs most, but in learned robot simulators this has not been tested against collecting where the policy goes. We provide that test with two tools: a controlled testbed in which seven acquisition rules, including an adversary given the model's true error, adapt the released NeRD models under identical data budgets and fine-tuning; and a paired-perturbation diagnostic that checks whether a model reproduces the engine's contact-event changes. The adversary finds errors 6–$12\times$ larger than random actions, yet yields policies no better than random data, whereas collecting along the trajectories of model-trained policies recovers 0.81–0.88 of the return of a policy trained in the engine on the jump task. Fine-tuning that cuts one-step error fivefold leaves most contact-event changes unreproduced, and the rule with the lowest boundary error trains worse policies than policy-trajectory collection. The value of adaptation data thus depends on whether the policy will visit its states, not on how wrong the model is there, and prediction accuracy, contact fidelity, and policy performance must be evaluated separately.

1 INTRODUCTION

Robot policies are increasingly trained not in a physics engine but in a neural network that has learned to imitate one: a *learned simulator* predicts the next state from recent states and torques, and because it runs as a batched forward pass and can be trained from data, policy optimization inside it is fast and cheap (Xu et al., 2025; Li et al., 2025a). The catch is that a policy trained inside a learned simulator does not learn the robot; it learns the simulator, mistakes included. On the publicly released NeRD model of the Ant robot, a jump policy scores $7.9\times$ higher in the model than in the physics engine the model was trained to imitate. This is the exploitation of model error familiar from model-based reinforcement learning (Kurutach et al., 2018; Rajeswaran et al., 2020), and it turns the question of how to train a learned simulator into the question of how to *adapt* one: given a pretrained model and a budget of new ground-truth transitions, which transitions should be collected?

The prevailing answer is to look for error. Adversarial data collection for world models seeks the transitions the model gets most wrong (Chen et al., 2023; Güzel et al., 2026), and recent theory of policy-aware simulator learning derives a sampler that maximizes one-step error under the true dynamics (Dann et al., 2026). The premise is that a large error is a valuable error. Yet one-step loss is not the objective that matters for control (Lambert et al., 2020), and a model needs to be accurate where a learner goes, not everywhere (Ross & Bagnell, 2012). The two principles have never been tested against each other on a learned robot simulator under matched data budgets and updates. Adaptation is still judged by held-out prediction error, and we show that this error can fall while control does not improve.

Our claim is that for adapting a simulator, *policy visitation matters more than error magnitude*: a new transition is useful because the policy we eventually train will visit that state, not because the model is wrong there. On the Ant model the two criteria pick different states. An adversary trained to find the model's largest errors keeps the robot crouched or tilted near the point of falling, in states

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.

the task policies trained afterwards rarely visit. Fine-tuning on its data reduces error in those states, but the policies rarely go there and perform no better. Contact adds a second gap. Near a contact boundary, where a small perturbation decides whether a foot touches the ground, the true dynamics change abruptly, and a model trained with a smooth one-step loss can match the next state closely yet miss the change in contact. Our second claim follows: *prediction accuracy, contact fidelity, and policy performance are separate axes of simulator quality*, and improvement on one does not guarantee improvement on the others.

Testing these claims calls for a comparison rather than a new algorithm: the question is which way of choosing data helps the policy, so everything else must be held fixed. All conditions start from the same released NeRD model, receive the same number of new transitions from the physics engine it was trained on, and undergo the same fine-tuning; only the source of the data differs. We deliberately give error-seeking its best chance: instead of estimating where the model is wrong, as a practical method would have to, our adversary is given the model's true error against the physics engine. Its competitors are collection from the states that model-trained policies visit, which is what our claim recommends; random actions, to show what data with no selection at all achieves; and collection at contact boundaries verified in the physics engine, to check whether aiming at contact directly is enough. Hypotheses and tests were fixed before the runs. For each adapted model we measure all three: (i) prediction accuracy, as one-step error on held-out states; (ii) contact fidelity, as whether the model reproduces the contact-event changes the physics engine shows under identical perturbations; and (iii) policy performance, as the return of policies trained inside it, evaluated in the physics engine and normalized by a policy trained there directly.

The outcome is one-sided. The adversary finds errors 6–$12\times$ larger than uniform data, yet policies trained in its adapted model perform no better than with random data, whereas collecting from policy-visited states recovers 0.81–0.88 of the directly trained policy's return on the jump task. The three axes come apart as predicted: fine-tuning that raises prediction accuracy at contact boundaries fivefold leaves contact fidelity almost unchanged, and the rule with the highest boundary accuracy trains among the worst policies.

Our contributions are as follows.

- **A controlled testbed for simulator adaptation.** We implement seven acquisition rules on public NeRD models, including an adversary given the true model error, and fine-tune all under identical budgets and schedules with pre-specified tests: the first head-to-head comparison of error-seeking and visitation-based acquisition on a learned robot simulator.
- **A contact-event diagnostic.** We locate verified contact boundaries in the physics engine and apply identical perturbations in engine and model to test whether the model reproduces the resulting contact changes, which one-step error does not measure.
- **Policy visitation beats error magnitude.** Error-seeking yields no policy improvement over random data even with oracle error; policy-visited data recover 0.81–0.88 of the directly trained return on jump. Lower prediction error did not by itself bring faithful contact events or better policies, so all three axes must be reported. We release protocols, data, and code.

2 Related work

Learned robot simulators. NeRD (Xu et al., 2025) trains a transformer to predict the next rigid-body state from engine trajectories under random torques and releases the models; the Robotic World Model (Li et al., 2025a;b) trains policies inside such models for real robots. The NeRD models studied here are pretrained on random-torque data and then reused; how to update a released model for the policies it will serve is the question we study.

Exploitation of model error. Policies optimized inside an imperfect model exploit its errors, a long-standing concern in model-based RL (Kurutach et al., 2018; Rajeswaran et al., 2020; Kidambi et al., 2020) and in proxy optimization more broadly (Sims, 1994; Koos et al., 2013; Muratore et al., 2021; Gao et al., 2023; Skalse et al., 2022). Bhamidipaty et al. (2026b) derive exploitation bounds and a safe horizon, MiraBench (Yang et al., 2026) measures optimism in robotic video world models, and ensembles and uncertainty penalties limit exploitation during policy learning (Chua et al., 2018; Clavera et al., 2018; Yu et al., 2020; Li et al., 2025b). These interventions constrain the policy; we instead ask how to change the model.

Which data to collect. Two principles compete. The first is error: disagreement-based exploration collects transitions on which an ensemble is uncertain (Pathak et al., 2019; Shyam et al., 2019; Sekar et al., 2020), GALILEO (Chen et al., 2023) fits offline dynamics models by adversarial reweighting, PROWL (Güzel et al., 2026) learns world models from adversarially collected trajectories, and RENEW (Bhamidipaty et al., 2026a) repairs a world model from preferences and uncertainty-selected transitions in grid worlds and classic-control tasks. The second is visitation: one-step prediction and control objectives are mismatched (Lambert et al., 2020), and the DAgger and agnostic-system-identification traditions fit policies and dynamics, respectively, on learner-visited states (Ross et al., 2011; Ross & Bagnell, 2012). Related work uses real-trial policy evaluation (Abbeel et al., 2006) or model-generated inputs (Venkatraman et al., 2015; Talvitie, 2014). Value-aware and value-equivalent objectives align models with downstream value computations (Farahmand et al., 2017; Grimm et al., 2020). To our knowledge, the two have not been compared on a pretrained neural simulator with contact under a matched budget and update, where they select different regions of state space. Rather than estimate prediction error, we give our adversary the exact normalized one-step prediction error.

Contact and learned dynamics. Contact makes dynamics nearly discontinuous: grazing impacts amplify small state changes (Nordmark, 1991), basin boundaries limit final-state predictability (Grebogi et al., 1983), stiffness hinders learning (Parmar et al., 2021) and first-order optimization (Suh et al., 2022b), and smoothing restores regularity at the cost of fidelity (Suh et al., 2022a; Pang et al., 2023). Graph-network and implicit models learn discontinuous contact (Allen et al., 2023; Pfrommer et al., 2021), Bianchini et al. (2022) study prediction loss near discontinuities, and learned weather models are known to suppress divergence that the true system exhibits (Selz & Craig, 2023). Our diagnostic measures not pointwise error but whether the model reproduces the change in contact sequence that the engine shows under an identical perturbation, at boundary states found by adversarial search (Lee et al., 2020; Uesato et al., 2018) with reference-state restarts (Kakade & Langford, 2002; Ecoffet et al., 2021) and verified under fresh perturbations.

Policy-aware simulator learning. PASL (Dann et al., 2026) bounds value error by model error under regularity and coverage conditions (see also Asadi et al., 2018) and, from its Error-MDP result, proposes a sampler that maximizes one-step error under the true dynamics, approximated with a learned critic, plus a task-constrained variant. Our EA uses exact normalized one-step error in GT, EA-M is a separate in-model adversary control, and our task-reward constraints are motivated by their constrained variant.

3 Experimental design

Figure 1 summarizes the protocol: seven acquisition rules adapt the released model under a common training budget and update schedule, differing only in which transitions they collect.

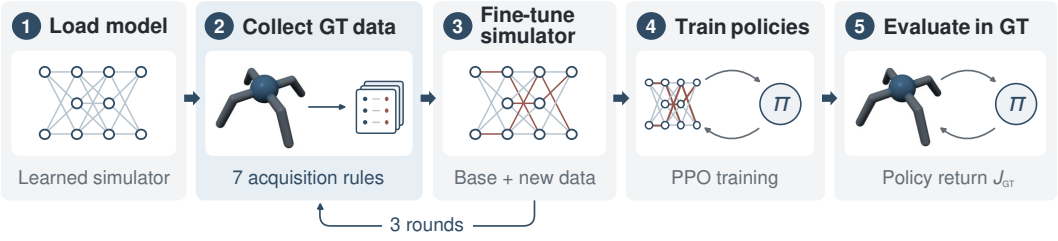


Figure 1: **Simulator adaptation and policy evaluation.** Every acquisition rule starts from the released NeRD checkpoint and shares the retained-data budget and fine-tuning schedule over three rounds. Prediction accuracy and contact fidelity are measured on each final model; policy performance is the GT return of fresh policies trained inside it.

3.1 Simulators, tasks, and policies

We use the public NeRD models (Xu et al., 2025) of Ant and ANYmal (Hutter et al., 2016); we call each original checkpoint the *released model*. NeRD predicts the next rigid-body state from a ten-

state history, joint torques, and contact features; its training data come from a GPU physics engine (Macklin, 2022), our *ground truth* (GT). Tasks are Ant run, Ant jump (our objective), ANYmal forward walk, and ANYmal jump, with 500-step episodes that end on falling.

Task policies are trained with PPO (Schulman et al., 2017) inside NeRD (*model-trained policies*) or in GT (*reference policies*); J_{GT} and J_{model} denote returns in GT and in the model, and J_{ref} the reference policy's GT return. The public training wrapper gives NeRD one state of context, which NeRD is trained for, since its loss supervises every position of the ten-state window, including the first; restoring the full context makes exploitation more frequent, not less (Appendix C).

The normalized one-step error of a model f at state s and action a is $e_f(s,a) = \|(f(s,a) - s') \oslash \mathrm{MAD}_\Delta\|_2$, where s' is the GT next state, $\oslash$ is elementwise division, and MAD_Δ is the per-coordinate median absolute deviation of one-step GT state changes along reference-policy trajectories, floored at 10^{-6}.

3.2 Contact-event diagnosis

Defining boundary states. We roll out a GT state and copies with slightly perturbed velocities under the same control rule and compare their *contact-event sequences*: which contacts occur and in what order, after short-lived contacts are filtered out. A *boundary state* is one whose sequences differ and whose return changes by more than a threshold within the rollout horizon; the threshold is set so that ordinary on-policy states rarely exceed it (0.26% false positives on an independent sample).

Finding and verifying boundary states. To find boundary states, we perturb states near reference-policy trajectories, sampling perturbations uniformly or with a cross-entropy search that focuses on the strongest contact and return changes. A state is kept only if the change recurs under new perturbation directions, and a synthetic case with a known contact change checks that the test detects it (Appendix A.2).

Measuring contact fidelity. At verified states, we apply identical perturbations in GT and the model under the same control rule and record how often, in each simulator, a perturbed rollout's contact-event sequence diverges from the unperturbed one. Contact fidelity compares the model's divergence rate with that of GT, a property one-step error does not measure.

3.3 Acquisition rules and budgets

The seven rules (Table 1) cover the options contrasted in §1: random actions (RA) as a no-selection baseline, an error-maximizing adversary (EA) for model error, boundary sampling (BS) for contact, and policy rollouts (PR) plus three visited-state rules (VS-U, VS-C, VS-V) for policy visitation. PR and the visited-state rules act with *exploiters*, three PPO policies trained for the task inside the current model and retrained after every round; EA instead trains an adversary in GT to maximize cumulative normalized one-step error.

Collection always takes place in GT. Policy-based rules use three policies trained in the current model for the task reward. These collection policies, also called exploiters, are refreshed after each adaptation round. The error-maximizing adversary instead learns in GT to maximize cumulative normalized one-step error. Resetting GT to a selected state allows repeated collection from intermediate points along a trajectory.

The visited-state rules pick $1{,}024$ of $4{,}096$ states visited by the exploiters in GT, perturb them, and collect from them. They differ only in how states are weighted: uniformly (VS-U), toward states that are both near a contact boundary and poorly predicted (VS-C), or toward states where the exploiter's critic values the GT and model next states differently (VS-V). The VS-C weight is the product of two terms: the fraction of perturbation directions passing the contact-and-return test at both tested scales (§3.2), and the model–GT state discrepancy after replaying the same ten actions in GT and the model.

Every rule starts from the released model and a shared base set of $1{,}024{,}000$ uniform-random GT transitions and adds $1{,}024{,}000$ transitions in each of three rounds. Each round fine-tunes the previous model on the base set plus all data collected so far, with identical updates and optimizer settings.

Table 1: The seven acquisition rules. Rules differ only in starting states, collection actions, and selection.

Rule	Starting states	Collection actions	Selection criterion
Random actions (RA)	Environment reset	Uniform random torques	None
Policy rollouts (PR)	Environment reset	Model policies + noise	Policy visitation
Error-maximizing adversary (EA)	Environment reset	GT-trained adversary	Cumulative one-step error
Boundary sampling (BS)	Verified boundary states	Uniform random torques	Contact/return sensitivity
Visited states, uniform (VS-U)	Model-policy visits in GT	Model policies + noise	Uniform over candidates
Visited states, contact-weighted (VS-C)	Same as VS-U	Same as VS-U	Contact sensitivity $\times$ model discrepancy
Visited states, value-weighted (VS-V)	Same as VS-U	Same as VS-U	Critic value disagreement

Budgets count retained transitions; GT steps spent on search and collector training are reported separately. The adversary spends about twenty such steps per retained transition, so matching retained data does not handicap error-seeking.

3.4 Measurements and statistics

Policy performance. In each final model we train two fresh PPO policies (*evaluation policies*) and report their mean $J_{\mathrm{GT}}/J_{\mathrm{ref}}$; a value of 1 matches training directly in GT.

Prediction accuracy. We report the median one-step error e_f under reference-policy actions on two held-out sets fixed before fine-tuning: *boundary states* (31 on Ant jump, 55 on Ant run; found away from every collection restart state), whose error we call *boundary error*, and *reference-policy states*. We also report it descriptively (without statistical tests) on a *policy-visited set* of states visited by the evaluation policies. Figures and tables show these errors relative to the released model, so values below 1 are improvements.

Contact fidelity. Descriptively, at the Ant jump boundary states, we run the diagnostic of §3.2 with zero torque and report each final model's divergence rate as a fraction of GT's; 1 means the model changes contacts as often as GT does.

Return overestimation. Each final model also scores a fixed panel of forty policies (seed 0 of every rule and run, plus five released-model policies). *Panel optimism* is the mean of $\max(0, \log_{10}(J_{\mathrm{model}}/J_{\mathrm{GT}}))$: 0 means no overestimation, and 10 to that power is the geometric-mean overestimation factor.

Replication and inference. Each rule has five independent runs. A pre-specified criterion limits formal inference to tasks where the released model leaves room for improvement (policy return $\leq 0.80\, J_{\mathrm{ref}}$): Ant jump (0.15) and run (0.70); ANYmal is descriptive. We use task-stratified permutation tests with Holm correction (Holm, 1979) and bootstrap 95% intervals (Agarwal et al., 2021). Primary contrasts compare VS-C, which combines visitation and contact, with EA on boundary error, policy return, and optimism; the twelve secondary contrasts are listed in Table 2. Metrics, thresholds, and contrasts were fixed before the runs (settings and later adjustments: Appendix A).

4 Policy optimization exploits task-dependent simulator error

Cross-evaluation. We train PPO policies for Ant run inside the released NeRD model or in GT and evaluate each in both (Fig. 2a). Over three seeds, the model over-estimates model-trained policies (model-minus-GT return $+185$) and under-estimates GT-trained ones (-1036). Model-trained

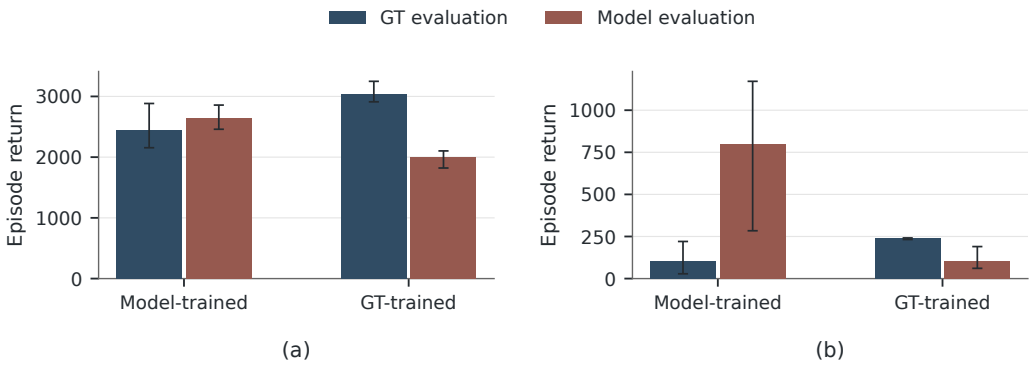


Figure 2: **Cross-evaluation of Ant policies.** (a) Ant run; (b) Ant jump. Policies are trained in the released model or GT and evaluated in both. Bars show mean episode return and whiskers the min–max over three training seeds.

policies also achieve lower GT return than GT-trained policies. In a separate 500-epoch experiment, the model-trained policies' evaluation gap widens from $+221$ at epoch 200 to $+356$ at epoch 500. This growing, policy-dependent gap is an operational signature of exploitation (Kurutach et al., 2018; Muratore et al., 2021); NeRD also reports overestimation on Ant run (Xu et al., 2025).

Task-induced distribution shift. Changing the task to jump, which NeRD's own evaluation does not cover, exposes much larger return overestimation in the same model (Fig. 2b). The ratio of mean model return to mean GT return rises from $1.08\times$ on run to $7.9\times$ on jump.

Robustness checks. The evaluation gap persists across implementation and context checks. Independent code and hardware reproduce all four learning-curve training/evaluation combinations within 3%. In a separate five-seed Ant jump check, preserving ten-step context increases extreme overestimation ($J_{\text{model}}/J_{\text{GT}} \geq 10^4$) from one seed to all five. Appendix C details these checks, fixed-policy re-evaluation, and a fix to the public Ant observation buffer applied in §6.

Exploitation thus depends on the task a policy optimizes, suggesting that the errors that matter lie along the policy's own trajectories. We next ask whether one-step prediction captures contact responses near those trajectories.

5 One-step error misses contact responses

We apply the contact-event diagnostic (§3.2) to the released model and to a model fine-tuned on its failures. These experiments use their own verified states and zero-torque rollouts; a state counts as divergent if any of four perturbation directions changes its contact-event sequence.

Verified boundary states are reproducible. Boundary states are rare near reference-policy trajectories (about 1% of candidates), but those found by targeted search are confirmed on retesting at 85% on Ant jump, and 87.5% in an independent repetition, against 0% for random controls.

The released model attenuates contact responses. At 34 verified Ant jump boundary states, perturbations change the contact-event sequence at 50.0% of states in GT but 2.9% in the model (McNemar $p = 1.4 \times 10^{-4}$; controls 2.2% vs. 0%; Fig. 9). Tenfold larger perturbations raise the model's rates to 20.6% at boundaries and 10.9% at controls, a boundary-to-control ratio of $1.9\times$ against $23\times$ in GT: larger perturbations make the model diverge at control states too, rather than specifically at boundaries.

Fine-tuning improves prediction but not contact fidelity. In a separate experiment, four cycles of fine-tuning on detected failure cases cut one-step error (zero-action probes) at 85 verified boundary states from 0.714 to 0.142, and equal-budget random data cut it to 0.245. Yet contact-event

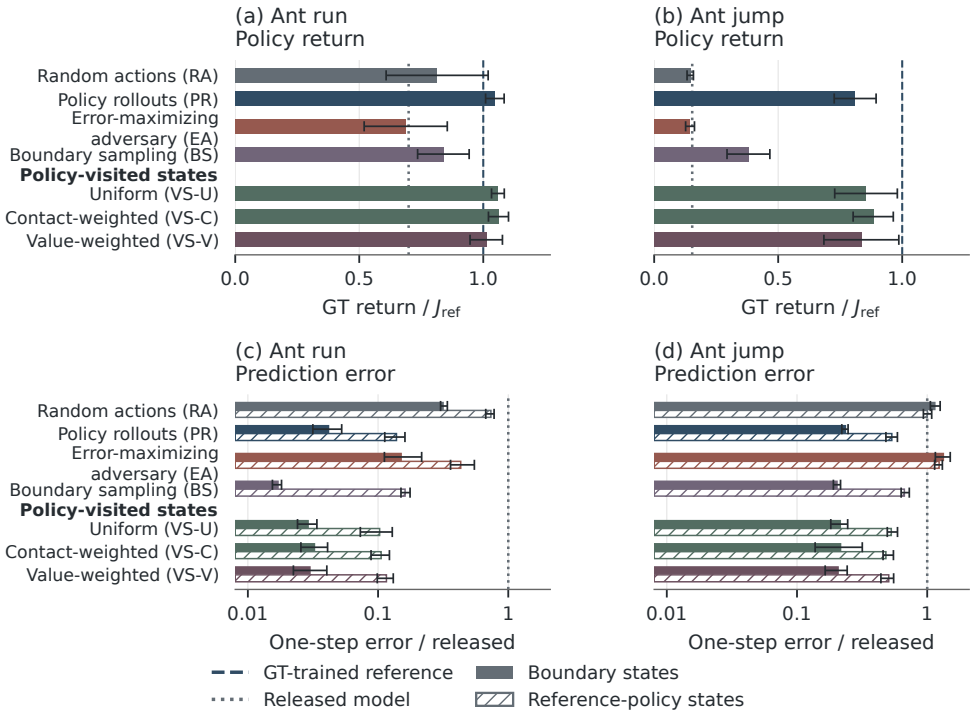


Figure 3: **Policy return and prediction error after acquisition.** (a,b) GT return $/J_{\text{ref}}$, averaged over two policy seeds per run; bars show the mean and whiskers ± 1 SD across five independent runs. (c,d) One-step error relative to the released model on boundary and reference-policy states, shown on logarithmic axes. Bar ends are geometric means over five runs; whiskers show min–max ranges. Strategies: Table 1; ANYmal results: Fig. 6.

divergence rises only from 4.7% to 7.1%, against 51.8% in GT (gap $+44.7$ pp $[+32.9, +56.5]$, $p = 4.1 \times 10^{-10}$), with no detected advantage over random data ($p = 1.0$). Better next-state prediction therefore does not bring faithful contact responses (details and sensitivity checks: Appendix C).

6 Policy Visitation Beats Error Magnitude

We first compare the seven rules (§6.1), then use follow-up controls to explain their differences (§6.2).

6.1 Main comparison

Visitation recovers control; boundary accuracy does not. On jump, PR and the three visited-state rules reach 0.81–$0.88\, J_{\text{ref}}$, against 0.14 for EA and 0.15 for RA (Fig. 3a,b), and cut boundary error to 0.21–0.24 of the released model's (Fig. 3c,d). BS attains the lowest boundary error on both tasks, yet its policies reach only $0.38\, J_{\text{ref}}$ on jump and trail every visitation rule on run. The visitation rules are also more accurate than BS on the policy-visited set of both tasks. Accuracy at boundary states therefore does not by itself determine policy performance.

Statistical tests confirm the ordering. VS-C beats EA on all three primary endpoints after Holm correction (Table 2), by $+0.74\, J_{\text{ref}}$ on jump and $+0.37\, J_{\text{ref}}$ on run; PR and BS also beat EA on return and optimism. Weighting within visited states is unresolved: all tested contrasts among VS-U, VS-C, and VS-V are inconclusive. On boundary error, the interval rules out an advantage of 0.5 SD or more for VS-C over BS (effect scales and power: Appendix A.3).

Table 2: **Statistical comparisons.** Ant jump and run; five runs per rule, two policy seeds per run. Task-stratified permutation tests, Holm-corrected within each family; effects and percentile-bootstrap 95% intervals are in between-run SD units (Appendix A). Negative effects favor the first rule for boundary error and optimism, positive effects for policy return. A negative conclusion excludes an advantage of 0.5 SD or more for the first rule; it does not establish equivalence.

Contrast	Endpoint	Holm p	Effect [95% CI]	Conclusion
Primary family (Holm-3)				
VS-C vs EA	boundary error	9×10^{-5}	$-14.5[-16.0, -13.0]$	positive (VS-C)
VS-C vs EA	policy return	9×10^{-5}	$+5.3[+4.6, +6.0]$	positive (VS-C)
VS-C vs EA	optimism	1.2×10^{-4}	$-6.2[-8.2, -4.4]$	positive (VS-C)
Secondary family (Holm-12)				
BS vs EA	policy return	0.011	$+1.9[+1.1, +2.7]$	positive (BS)
BS vs EA	optimism	0.0019	$-5.8[-8.1, -3.6]$	positive (BS)
BS vs EA	boundary error	3.6×10^{-4}	$-12.3[-13.0, -11.6]$	positive (BS)
VS-C vs VS-U	policy return	1	$+0.2[-0.5, +0.9]$	inconclusive
VS-C vs VS-U	optimism	1	$-0.0[-0.7, +0.6]$	inconclusive
VS-C vs VS-U	boundary error	1	$+0.3[-1.1, +1.7]$	inconclusive
VS-V vs VS-U	boundary error	1	$-0.0[-0.8, +0.8]$	inconclusive
VS-C vs VS-V	boundary error	1	$+0.5[-1.1, +2.0]$	inconclusive
EA vs PR	policy return	5×10^{-4}	$-4.9[-5.6, -4.2]$	positive (PR)
EA vs PR	optimism	0.0013	$+5.9[+4.2, +7.8]$	positive (PR)
VS-C vs BS	boundary error	0.073	$+2.0[+0.9, +3.1]$	negative for VS-C
BS vs RA	boundary error	4.4×10^{-4}	$-12.4[-12.7, -12.2]$	positive (BS)

Visitation-adapted models also evaluate policies accurately. On jump, all forty (run, seed) outcomes of the visitation rules exceed all twenty RA and EA outcomes. Their models cut panel optimism from 1.605 to 0.016–0.018 and over-estimate no panel policy by more than $1.4\times$, even policies that the models they were trained in over-estimate by up to $10^9\times$. They also rank the panel almost perfectly (Spearman 0.98, against 0.05 for EA). Full prediction, optimism, cost, and ANYmal results are in Appendix B.

Policy recovery does not restore contact fidelity. Final-model divergence rates stay at 7–19% of GT for all seven rules, including those whose policies recover (Table 9). Because these rates count perturbed rollouts rather than states, and use different states and perturbations, they are not comparable with those of §5.

Released-model boundary error is $1.1\times$ reference-policy error on jump, versus $3.6\times$ on run. The boundary-error increase is therefore task-dependent, whereas §5 examines discrepancies in contact responses. Appendix B reports absolute errors on all three evaluation sets.

6.2 Follow-up controls distinguish task relevance, actions, and forgetting

We use additional Ant jump experiments to test task relevance, collection actions, and forgetting. These follow-ups are descriptive. Repeated PR, EA, RA, and BS baselines reproduce the main study's qualitative ordering and approximate return levels (Appendix B.4).

Error maximization and policy-state proximity. In the main study, EA achieves its collection objective: released-model error on its data is 6–12$\times$ that on uniform data for Ant. Yet few collected states lie near states visited by evaluation policies in MAD-scaled velocity space: 9% on jump and 0% on run, versus 37–40% and 17–24% for the visitation strategies (exploratory; calibrated radius in Appendix B.2). The adversary often crouches or tilts near termination. In the follow-up batch, prediction improves on a held-out set collected by released-model adversaries, while policy return remains low (Table 12). Thus improving prediction on the adversary distribution does not yield comparable control gains.

Task constraints and adversary initialization. We constrain mean step task reward to 50% or 90% of the reference-policy value. Cold starts use random initialization; the warm start uses the reference policy at the 90% floor. At this matched floor, warm starting raises return from 0.213 to $0.565\, J_{\mathrm{ref}}$ (warm-start between-run SD 0.178, three runs); the lower-floor cold start reaches 0.187. Constraint violations fall from all training epochs to 53–66% (Table 13). Return remains below

PR and reference-policy collection, which replaces VS-U's exploiters with the frozen GT-trained reference policy (Table 12). Fixed-coefficient task-reward augmentation and in-model search remain near the low-return baselines (Table 10).

Policy-rollout mixtures. We next vary the amount of task-relevant data directly by replacing fractions of PR's data with EA's. Return decreases as the adversary fraction rises, but retaining a quarter of the budget for policy rollouts yields $0.697\,J_{\mathrm{ref}}$, or 78% of PR's return (Table 12). EA model rollouts exhibit spurious sustained flight (Fig. 5), which is absent from mixture-model rollouts.

Boundary-error regression under fine-tuning alone. The no-new-data control uses only the original base set with the same fine-tuning schedule. Its jump boundary error reaches 1.310 times the released model's boundary error, reproducing much of the regression under RA and EA. Forgetting induced by fine-tuning therefore contributes to that regression; it cannot be assigned solely to the acquired data.

Start states and collection actions. At matched boundary starts, switching uniform torques (BS) to exploiter actions raises return from 0.390 to $0.799\,J_{\mathrm{ref}}$. At policy-visited starts, uniform torques also support recovery, reaching $0.771\,J_{\mathrm{ref}}$ (between-run SD 0.131, three runs). The matched-batch comparison of random and policy actions at visited starts is incomplete (Appendix B.7).

Policy selection across acquisition conditions. The follow-up panel contains one evaluation policy per condition and run (13×3), plus five released-model policies. Each adapted model excludes its condition's three policies and selects among the remaining 41 by predicted return. GT-return regret is the fraction lost relative to the best remaining candidate. Regrets are 9.4% for policy rollouts, 0% for boundary starts with model-policy actions, and 18.2%/20.1% for reference-policy collection and the warm-started adversary. The latter two have high Spearman correlations between model and GT returns (0.932/0.941), supporting evaluation across conditions. Appendix B.6 compares these candidate sets with the full 44-policy panel.

7 Conclusion and Limitations

Model error tells us where a learned simulator is wrong; policy visitation tells us where it needs to be right. Under matched budgets, data gathered where the policy goes trained policies close to direct engine training, while data gathered where the model erred most, even with exact error, did no better than random. The same gap appears in evaluation: a simulator can predict the next state well and still misrepresent what contact does next. We suggest adapting simulators where their policies go and judging them by the policies they train and rank, not by one-step loss alone; contact-specific inductive biases (Allen et al., 2023; Bianchini et al., 2022) may close the remaining contact gap.

Limitations. Formal evidence comes from one simulator family, one learner (PPO), and two Ant tasks; ANYmal is descriptive. The ground truth permits state resets and exact-error queries, so matching retained transitions measures the utility of training data, not the cost of interaction on a physical robot. The contact diagnostic probes finite-horizon responses at chosen perturbation scales; the mismatch it reveals does not establish a discontinuity or its causal role in exploitation.

REPRODUCIBILITY STATEMENT

The models, environments, and reference physics engine are public. Appendix A specifies acquisition procedures, fine-tuning and PPO settings, evaluation sets, and statistical analysis. Appendix B provides the full results and additional controls; Appendix C describes the exploitation and contact-boundary experiments. Saved measurements support recomputation of the acquisition tables without retraining, and the manuscript source includes the numerical inputs and plotting code for its graphs.

The main acquisition study used approximately 538 GPU-hours and 7.8×10^9 GT steps, including 4.8×10^8 retained fine-tuning transitions. The task-reward and in-model adversary experiments added about 19 GPU-hours; the additional acquisition controls used approximately 72.8 GPU-hours on A100 and H100 GPUs. Appendix B.7 reports the completed measurements from the action comparison at policy-visited starts.

AI USE STATEMENT

We used generative AI tools to assist with manuscript editing and translation, identifying relevant literature, and implementing and debugging code for the experimental setup. The authors reviewed the resulting text, checked references against the original publications, and tested the AI-assisted code. The authors take responsibility for the final manuscript, implementation, and reported results.

REFERENCES

Pieter Abbeel, Morgan Quigley, and Andrew Y. Ng. Using inaccurate models in reinforcement learning. In *International Conference on Machine Learning (ICML)*, 2006.

Rishabh Agarwal, Max Schwarzer, Pablo Samuel Castro, Aaron Courville, and Marc G. Bellemare. Deep reinforcement learning at the edge of the statistical precipice. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2021.

Kelsey R. Allen, Tatiana Lopez Guevara, Yulia Rubanova, Kim Stachenfeld, Alvaro Sanchez-Gonzalez, Peter Battaglia, and Tobias Pfaff. Graph network simulators can learn discontinuous, rigid contact dynamics. In *Conference on Robot Learning (CoRL), PMLR 205*, 2023.

Kavosh Asadi, Dipendra Misra, and Michael L. Littman. Lipschitz continuity in model-based reinforcement learning. In *International Conference on Machine Learning (ICML), PMLR 80*, 2018.

Logan Mondal Bhamidipaty, Mykel J. Kochenderfer, and Subramanian Ramamoorthy. RENEW: Towards learning world models and repairing model exploitation from preferences. *arXiv preprint arXiv:2607.14180*, 2026a.

Logan Mondal Bhamidipaty, Esmeralda S. Whitammer, David Abel, Mykel J. Kochenderfer, and Subramanian Ramamoorthy. Imperfect world models are exploitable. *arXiv preprint arXiv:2605.15960*, 2026b.

Bibit Bianchini, Mathew Halm, Nikolai Matni, and Michael Posa. Generalization bounded implicit learning of nearly discontinuous functions. In *Learning for Dynamics and Control (L4DC), PMLR 168*, 2022.

Xiong-Hui Chen, Yang Yu, Zhengmao Zhu, Zhihua Yu, Zhenjun Chen, Chenghe Wang, Yinan Wu, Hongqiu Wu, Rong-Jun Qin, Ruijin Ding, and Fangsheng Huang. Adversarial counterfactual environment model learning. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2023.

Kurtland Chua, Roberto Calandra, Rowan McAllister, and Sergey Levine. Deep reinforcement learning in a handful of trials using probabilistic dynamics models. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2018.

Ignasi Clavera, Jonas Rothfuss, John Schulman, Yasuhiro Fujita, Tamim Asfour, and Pieter Abbeel. Model-based reinforcement learning via meta-policy optimization. In *Conference on Robot Learning (CoRL)*, 2018.

Christoph Dann, Yishay Mansour, and Mehryar Mohri. Theoretical foundations and effective algorithms for policy-aware simulator learning. *arXiv preprint arXiv:2605.29032*, 2026.

Adrien Ecoffet, Joost Huizinga, Joel Lehman, Kenneth O. Stanley, and Jeff Clune. First return, then explore. *Nature*, 590:580–586, 2021.

Amir-massoud Farahmand, André Barreto, and Daniel Nikovski. Value-aware loss function for model-based reinforcement learning. In *Artificial Intelligence and Statistics (AISTATS), PMLR 54*, 2017.

Leo Gao, John Schulman, and Jacob Hilton. Scaling laws for reward model overoptimization. In *International Conference on Machine Learning (ICML), PMLR 202*, 2023.

Celso Grebogi, Steven W. McDonald, Edward Ott, and James A. Yorke. Final state sensitivity: An obstruction to predictability. *Physics Letters A*, 99(9):415–418, 1983.

Christopher Grimm, André Barreto, Satinder Singh, and David Silver. The value equivalence principle for model-based reinforcement learning. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2020.

Ahmet H. Güzel, Jenny Seidenschwarz, Benjamin Graham, Jonathan Sadeghi, Jeffrey Hawke, and Ilija Bogunovic. PROWL: Prioritized regret-driven optimization for world model learning. *arXiv preprint arXiv:2605.18803*, 2026.

Sture Holm. A simple sequentially rejective multiple test procedure. *Scandinavian Journal of Statistics*, 6(2):65–70, 1979.

Marco Hutter, Christian Gehring, Dominic Jud, Andreas Lauber, C. Dario Bellicoso, Vassilios Tsounis, Jemin Hwangbo, Karen Bodie, Peter Fankhauser, Michael Bloesch, Remo Diethelm, Samuel Bachmann, Amir Melzer, and Mark Hoepflinger. ANYmal—a highly mobile and dynamic quadrupedal robot. In *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pp. 38–44, 2016. doi: 10.1109/IROS.2016.7758092.

Sham Kakade and John Langford. Approximately optimal approximate reinforcement learning. In *International Conference on Machine Learning (ICML)*, 2002.

Rahul Kidambi, Aravind Rajeswaran, Praneeth Netrapalli, and Thorsten Joachims. MOReL: Model-based offline reinforcement learning. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2020.

Sylvain Koos, Jean-Baptiste Mouret, and Stéphane Doncieux. The transferability approach: Crossing the reality gap in evolutionary robotics. *IEEE Transactions on Evolutionary Computation*, 17 (1):122–145, 2013.

Thanard Kurutach, Ignasi Clavera, Yan Duan, Aviv Tamar, and Pieter Abbeel. Model-ensemble trust-region policy optimization. In *International Conference on Learning Representations (ICLR)*, 2018.

Nathan Lambert, Brandon Amos, Omry Yadan, and Roberto Calandra. Objective mismatch in model-based reinforcement learning. In *Learning for Dynamics and Control (L4DC), PMLR 120*, 2020.

Ritchie Lee, Ole J. Mengshoel, Anshu Saksena, Ryan W. Gardner, Daniel Genin, Joshua Silbermann, Michael Owen, and Mykel J. Kochenderfer. Adaptive stress testing: Finding likely failure events with reinforcement learning. *Journal of Artificial Intelligence Research*, 69:1165–1201, 2020.

Chenhao Li, Andreas Krause, and Marco Hutter. Robotic world model: A neural network simulator for robust policy optimization in robotics. *arXiv preprint arXiv:2501.10100*, 2025a.

Chenhao Li, Andreas Krause, and Marco Hutter. Uncertainty-aware robotic world model makes offline model-based reinforcement learning work on real robots. *arXiv preprint arXiv:2504.16680*, 2025b.

Miles Macklin. Warp: A high-performance python framework for GPU simulation and graphics, 2022. NVIDIA GPU Technology Conference; https://github.com/nvidia/warp.

Fabio Muratore, Michael Gienger, and Jan Peters. Assessing transferability from simulation to reality for reinforcement learning. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 43(4), 2021.

Arne B. Nordmark. Non-periodic motion caused by grazing incidence in an impact oscillator. *Journal of Sound and Vibration*, 145(2):279–297, 1991.

Tao Pang, H. J. Terry Suh, Lujie Yang, and Russ Tedrake. Global planning for contact-rich manipulation via local smoothing of quasi-dynamic contact models. *IEEE Transactions on Robotics*, 39 (6):4691–4711, 2023.

Mihir Parmar, Mathew Halm, and Michael Posa. Fundamental challenges in deep learning for stiff contact dynamics. In *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pp. 5181–5188, 2021.

Deepak Pathak, Dhiraj Gandhi, and Abhinav Gupta. Self-supervised exploration via disagreement. In *International Conference on Machine Learning (ICML)*, 2019.

Samuel Pfrommer, Mathew Halm, and Michael Posa. ContactNets: Learning discontinuous contact dynamics with smooth, implicit representations. In *Conference on Robot Learning (CoRL), PMLR 155*, 2021.

Aravind Rajeswaran, Igor Mordatch, and Vikash Kumar. A game theoretic framework for model based reinforcement learning. In *International Conference on Machine Learning (ICML), PMLR 119*, 2020.

Stéphane Ross and J. Andrew Bagnell. Agnostic system identification for model-based reinforcement learning. In *International Conference on Machine Learning (ICML)*, 2012.

Stéphane Ross, Geoffrey J. Gordon, and J. Andrew Bagnell. A reduction of imitation learning and structured prediction to no-regret online learning. In *Artificial Intelligence and Statistics (AISTATS)*, 2011.

John Schulman, Filip Wolski, Prafulla Dhariwal, Alec Radford, and Oleg Klimov. Proximal policy optimization algorithms. *arXiv preprint arXiv:1707.06347*, 2017.

Ramanan Sekar, Oleh Rybkin, Kostas Daniilidis, Pieter Abbeel, Danijar Hafner, and Deepak Pathak. Planning to explore via self-supervised world models. In *International Conference on Machine Learning (ICML)*, 2020.

Tobias Selz and George C. Craig. Can artificial intelligence-based weather prediction models simulate the butterfly effect? *Geophysical Research Letters*, 50, 2023. e2023GL105747.

Pranav Shyam, Wojciech Jaśkowski, and Faustino Gomez. Model-based active exploration. In *International Conference on Machine Learning (ICML)*, 2019.

Karl Sims. Evolving virtual creatures. In *SIGGRAPH*, pp. 15–22, 1994.

Joar Skalse, Nikolaus H. R. Howe, Dmitrii Krasheninnikov, and David Krueger. Defining and characterizing reward hacking. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2022.

H. J. Terry Suh, Tao Pang, and Russ Tedrake. Bundled gradients through contact via randomized smoothing. *IEEE Robotics and Automation Letters*, 7(2):4000–4007, 2022a.

H. J. Terry Suh, Max Simchowitz, Kaiqing Zhang, and Russ Tedrake. Do differentiable simulators give better policy gradients? In *International Conference on Machine Learning (ICML), PMLR 162*, 2022b.

Erik Talvitie. Model regularization for stable sample rollouts. In *Uncertainty in Artificial Intelligence (UAI)*, 2014.

Jonathan Uesato, Ananya Kumar, Csaba Szepesvári, Tom Erez, Avraham Ruderman, Keith Anderson, Krishnamurthy Dvijotham, Nicolas Heess, and Pushmeet Kohli. Rigorous agent evaluation: An adversarial approach to uncover catastrophic failures. *arXiv preprint arXiv:1812.01647*, 2018.

Arun Venkatraman, Martial Hebert, and J. Andrew Bagnell. Improving multi-step prediction of learned time series models. In *AAAI Conference on Artificial Intelligence*, 2015.

Jie Xu, Eric Heiden, Iretiayo Akinola, Dieter Fox, Miles Macklin, and Yashraj Narang. Neural robot dynamics. In *Conference on Robot Learning (CoRL)*, 2025. arXiv:2508.15755.

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.

Tianhe Yu, Garrett Thomas, Lantao Yu, Stefano Ermon, James Zou, Sergey Levine, Chelsea Finn, and Tengyu Ma. MOPO: Model-based offline policy optimization. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2020.

A Experimental design of the acquisition comparison

Experimental design. The primary contrast is VS-C versus EA. Each task and condition uses five independent acquisition runs, with search, exploiter training, adversary training, and collection RNG keyed by run index. Two PPO seeds are averaged into each run's policy-performance cell. A task inclusion criterion applied before adaptation excludes tasks without room for improvement from formal inference while retaining their measurements. Budgets count both retained training transitions and total GT steps. All conditions retain the released deterministic transition model.

A.1 Shared setup and acquisition procedures

Simulator and task settings. Policies use PPO with $2{,}048$ parallel environments. Both robots use the source engine's Featherstone integrator with compliant contact, with 16 substeps for Ant and 10 for ANYmal. NeRD pretraining rolls out sampled initial states under uniform random torques; for Ant, trunk height is 1.0–1.3 m at arbitrary orientation. The Ant jump objective introduced here is $3e^{2(\max(h,h_0)-h_0)} + \max(0, v_\uparrow)$, where h is body height, $h_0 = 0.6\,\text{m}$ and $v_\uparrow$ is centre-of-mass (CoM) vertical velocity. Episodes last 500 steps and terminate on falling; for Ant, the termination height is $0.3\,\text{m}$. Earlier jump measurements retain their original protocols and matched baselines (Appendix C).



Figure 4: **Robot models.** The Ant and ANYmal simulation models used in the experiments, shown at independent scales.

Common protocol. Free-joint velocities are represented as angular and centre-of-mass velocities so that physical perturbations are invariant to horizontal translation. Each condition starts from the released checkpoint and receives three rounds of $1{,}024{,}000$ new GT transitions, alongside a shared base set of $1{,}024{,}000$ uniform-random transitions per task. Collection uses GT-only engine instances and $T = 150$-step trajectories truncated at termination. ANYmal PD gains are fixed at $K_p = 85$, $K_d = 2$ for every condition. Each round uses 7 trainer epochs (6,000 iterations), batch size 512, sequence length 10, learning rate $10^{-4} \to 2 \times 10^{-5}$, and gradient norm 1. Fine-tuning starts from the preceding round's final model and uses the base set together with that condition's accumulated rounds $1..r$.

Policy rollouts and candidate states. For PR, VS-U, VS-C and VS-V, the exploiter ensemble comprises three PPO policies trained for 500 epochs each inside the current condition's model (round 1 shares one ensemble across the four conditions since all start from the released model); their GT rollouts (512 environments each) are the PR data and the candidate pool for the strategies that collect from teleported states (states every 5 steps, sub-sampled to $4{,}096$, re-centred to $x = z = 0$).

Selection and collection at visited states. Contact weighting (VS-C) uses

$$c(s) = \frac{1}{16} \sum_{d=1}^{16} \prod_{\sigma \in \{10^{-4}, 10^{-3}\}} b(s, d, \sigma), \qquad w_{\mathrm{C}}(s) = c(s)\, \widetilde{d}_{10}(s). \tag{1}$$

Here $b(s, d, \sigma) = 1$ when a valid closed-loop probe under the exploiter at state s, direction d, and scale σ exhibits both contact-event divergence and a return difference reaching the calibrated threshold; otherwise it is zero. Probes run for 145 frames. A direction contributes only if both scales pass, and states failing the unperturbed-twin check receive $c(s) = 0$. For $\widetilde{d}_{10}$, we replay the same ten exploiter actions in GT and the model, measure the normalized state error at step 10, and divide by its candidate-pool median (floored at 10^{-9}). Value weighting (VS-V) similarly normalizes $|\hat{V}(o(s'_{\mathrm{GT}})) - \hat{V}(o(s'_{\mathrm{model}}))|$ by its pool median, where o maps states to policy observations and $\hat{V}$ is the exploiter critic. We select $1{,}024$ sources by weighted sampling without replacement, using weights $w + 0.01 \max w$; all-zero weights and VS-U give uniform sampling. Collection resets GT to each source, applies log-uniform perturbations $\sigma \in [10^{-4}, 3 \times 10^{-3}]$, and runs ten trajectories per source with exploiter actions plus $\mathcal{N}(0, 0.2^2)$ noise.

Error-maximizing acquisition. For EA, PPO trains the adversary in GT with 2,048 environments, horizon 32 and 300 epochs per round. Episodes last 100 steps with the task's own termination (task episodes last 500 steps; collection trajectories 150). The reward is $\|(f_{\text{model}}(s, a) - f_{\text{GT}}(s, a)) \oslash \mathrm{MAD}_{\Delta}\|_2$ evaluated by a lock-step model instance (capped at 100 after a numerical blow-up on ANYmal; the cap bound 0.2% of adversary steps study-wide, 0–2.6% across individual task/run/round cells, and the uncapped exact error is what every reported measurement uses); collection from resets with the stochastic adversary. Dann et al. (2026)'s Algorithm 3 trains its adversary inside the learned simulator against a critic-estimated error; ours runs in the ground truth with the exact error, which follows their Error-MDP duality (their Thm. 9) rather than its practical approximation; the exact reward is a cleaner signal, but a ground-truth adversary is confined to physically reachable states, whereas an in-simulator adversary could enter the model's hallucinated regime; the follow-up in-model adversary (EA-M) (Appendix B) tests exactly this. The efficacy check requires the ratio of mean one-step error in EA round-1 data to that in RA data to exceed 2. Its outcome and collection-state diagnostics are reported in Appendix B.2.

Boundary sampling. For each BS run, we harvest $4{,}096$ reference-policy states and run cross-entropy search on the perturbation shell ($2{,}048 \times 18$ probes). The archive groups detections by the contact-site group that first diverges, divergence-time bin, perturbation-scale decade, and initial ground-contact status. Every archive cell is retested in 16 fresh directions. Collection then resets GT to verified sources and applies perturbations followed by uniform torques.

A.2 Diagnostics and held-out evaluation sets

Paired-perturbation protocol. The measurements in the main text use $\sigma \in [10^{-4}, 10^{-2}]$ velocity-MAD units and 4–16 perturbation directions, with the specific settings fixed by each experiment. Velocities are scaled by their per-coordinate MAD along reference-policy GT trajectories; prediction errors instead use the MAD of one-step state changes. We denote these two scales by MAD_v and MAD_{Δ}, respectively. The reference-policy action track evaluates each held-out state with the frozen reference policy's action; the zero-action track uses zero torques. Re-centering translates horizontal position to $x = z = 0$ while preserving physical velocity. Debouncing removes short contact episodes before event-sequence comparison. A valid boundary detection must recur under fresh perturbation directions and the diagnostic must detect a synthetic positive control with a known change in contact-event patterns. Severity is the return difference divided by its calibrated detection threshold, evaluated at the smallest passing perturbation scale and maximized over tested

directions. The calibration and its task-specific outcomes follow below; individual earlier protocols are in Appendix C.

Instrument calibration (per task). The return-divergence multiplier c_D is twice the 99.9th percentile ($p_{99.9}$) of normalized return differences in the on-policy null sample at the smallest probe scale, $\sigma_{\min}$. It sets the return threshold used together with contact-event divergence; false-positive rate verified on an independent sample (0.26% on all four tasks); synthetic positive control (a known change in contact-event patterns during mid-air roll-spin) fired on the three Ant/ANYmal-walk tasks (severity 1.9–3.4) and not on ANYmal jump, whose BS and VS-C conditions therefore fail the positive-control calibration of this diagnostic (descriptive task anyway).

Evaluation sets. The boundary set uses an independent harvest, uniform shell sampling, GT verification, 16-direction retesting, and ϵ-ball exclusion around every teleport source of every condition and run (31/55/28/21 verified states on Ant jump / Ant run / ANYmal walk / ANYmal jump). The reference-policy set contains 64 states from an independent reference-policy rollout seed. The descriptive policy-visited set pools GT rollouts of all seventy evaluation policies (16 environments each) and samples 64 states uniformly. Its construction rule was fixed before execution, but the states are generated after policy training.

A.3 Statistical analysis and interpretation rules

Budget-averaged error. For each run, let f_r and B_r denote the model and cumulative budget after round r. Trapezoidal integration gives the budget-averaged log error ratio $\bar{\ell}_B$:

$$\bar{\ell}_B = \frac{1}{B_3} \sum_{r=1}^{3} \frac{\ell_{r-1} + \ell_r}{2} (B_r - B_{r-1}), \qquad \ell_r = \log_{10} R_{\text{boundary}}(f_r), \quad B_0 = \ell_0 = 0. \tag{2}$$

Here $R_{\text{boundary}}(f_r)$ is the median boundary-set one-step error of model f_r divided by that of the released model, using the reference-policy actions of §3.4. Budget counts retained training transitions or total GT steps including search. Normalization by B_3 yields the mean log error ratio along each run's budget axis; lower is better. Absolute costs are reported separately.

Statistics. Cell value per (task, condition, run): prediction error $= \log_{10}$ median one-step error on the boundary set (reference-policy action track, final round-3 model); policy return = mean over the two seeds of $J_{\text{GT}}/J_{\text{ref}}$; budget-averaged error as in Eq. 2; panel optimism as defined in §3.4. Test statistic $S = \sum_c (\bar{Y}_{c,a} - \bar{Y}_{c,b})/s_c$ where c indexes tasks, a, b are the contrasted conditions, and $\bar{Y}_{c,a}$ is the mean endpoint cell value over the five runs of condition a on task c; s_c is the pooled between-run SD of the five conditions not in the contrast (root mean of their sample variances); null by shuffling the a/b labels among the ten runs within each task (10^5 Monte-Carlo permutations, two-sided, as specified in the design); effect S/C with $C = 2$ the number of tasks; CI by percentile bootstrap over runs within tasks (10^4). Primary family Holm-3; secondary family of twelve contrasts Holm-12; conclusions are defined below. Power (simulation, 300 trials): 0.42–0.54 for a 1 SD effect in the two-task analysis set unadjusted, 0.27–0.35 after Holm-3; 80% MDE ≈ 1.5 SD in this set (0.8 SD had all four tasks entered). *Verification.* The stratified null has $\binom{10}{5}^2 = 63{,}504$ equally likely label arrangements; enumerating all of them gives exact two-sided $p = 3.1 \times 10^{-5}$ (the smallest attainable value) for the five contrasts whose Monte-Carlo p lies at the resolution floor (VS-C vs EA on prediction error and policy return, BS vs EA on prediction error, BS vs RA on prediction error, EA vs PR on policy return), 9.4×10^{-5} for VS-C vs EA on panel optimism, 1.6×10^{-4} and 1.3×10^{-4} for BS vs EA and EA vs PR on panel optimism, 1.7×10^{-3} for BS vs EA on policy return and 1.2×10^{-2} for VS-C vs BS on prediction error; no conclusion changes. *Scale of the effects.* The pooled SDs are prediction error 0.042/0.065 $\log_{10}$, policy return 0.103/0.109 J_{ref}, panel optimism 0.063/0.0048 $\log_{10}$ (jump/run); the per-task standardized VS-C$-$EA differences are -18.7/-10.2 (prediction error), $+7.1$/$+3.4$ (policy return) and -9.7/-2.8 (panel optimism), the last run-task value being an absolute difference of 0.013 in $\log_{10}$ optimism on a task whose released model has none to remove. Because each cell already averages seeds, episodes or hold-out states, this unit is a repeatability scale: SD values express how reliably an ordering reproduces across independent runs and are not comparable to Cohen's-d conventions. No cell was excluded (the divergence rule was never triggered in the analysis set). Within a run, PR, VS-U, VS-C and

VS-V share the round-1 exploiter ensemble and VS-U, VS-C, VS-V the candidate pool, so their cells are positively dependent; the two-sample permutation test is valid under the exchangeability null but conservative for these clean contrasts.

Contrast families. Primary (Holm-3): VS-C vs EA on prediction error, policy return, panel optimism. Secondary (Holm-12): BS vs EA on policy return, panel optimism, prediction error; VS-C vs VS-U on policy return, panel optimism, prediction error; VS-V vs VS-U on prediction error; VS-C vs VS-V on prediction error; EA vs PR on policy return, panel optimism; VS-C vs BS on prediction error; BS vs RA on prediction error. Bootstrap: runs resampled with replacement within each (task, condition), the pooled SD fixed at its observed value.

Decision rules. Conclusions: *positive* (Holm $p < 0.05$); *negative* (Holm $p \geq 0.05$ and the 95% interval excludes an advantage of 0.5 SD or more for the first-named condition—a one-sided statement about that condition, not equivalence); *inconclusive (under-powered)* otherwise, reported with the minimum detectable effect.

A.4 Interpretation and protocol adjustments

Interpretation. The main contrasts compare complete acquisition procedures. Only the weighting contrasts hold the remaining procedure fixed. We interpret the VS-C–VS-U result through its confidence interval; a non-significant difference does not establish equivalence or that weighting is unnecessary. Prediction error, contact-event fidelity, and policy performance are separate endpoints. The policy-selection panels contain policies generated by the study; Appendix B.6 evaluates selection after excluding policies trained under the same acquisition condition.

Evaluation-panel adjustment. Before any panel optimism measurements, the panel was reduced from seventy evaluation policies plus five released-model policies evaluated over 2,048 episodes each to thirty-five seed-0 evaluation policies plus five released-model policies over 256 episodes each. The optimism metric and decision rule were unchanged. No main-study metric definition or decision threshold changed during execution; the adversary reward cap is specified above.

Implementation and completeness. All primary-analysis cells are complete. Two descriptive ANYmal runs required partial re-collection with the same exploiters, candidate pools, and boundary sources; the primary analysis was unaffected. The executed ANYmal integrator was Featherstone with 10 substeps throughout; an earlier description of XPBD was corrected without changing the simulations. Error ratios always use the released model re-measured in the same call.

Additional controls. The control experiments in Appendix B.4 were specified before their endpoint measurements. The warm-started adversary was added after observing the cold-started adversary's first-round dual-variable diagnostic and before any policy return or prediction error result in that batch. The reference reward used for constraint floors was corrected to use complete episodes before collection; the baseline-relative interpretation thresholds were specified before the new conditions' policy return results. These experiments are descriptive and do not alter the main statistical contrasts. The action comparison in Appendix B.7 requires three completed runs of each condition; the incomplete paired comparison is reported without a formal conclusion.

B Full result tables of the acquisition comparison

We first report the seven-strategy comparison, then collection diagnostics and follow-up controls. Policy-selection results are grouped with the analysis excluding policies trained under the same condition; incomplete comparisons of actions at policy-visited starts are reported last.

B.1 Main experimental results

All values are five-run means over the final (round-3) models unless noted. Ratios are relative to the released model re-evaluated in the same call (prediction error) or to the GT-trained reference policy (policy return).

Table 3: Policy return: GT return of the policy trained inside each final model, $/J_{\mathrm{ref}}$ (mean of 2 seeds per run, then mean over runs). Released-model policies: Ant jump 0.15, Ant run 0.70, ANYmal walk 1.03, ANYmal jump 1.01. ANYmal rows are descriptive (runs in parentheses).

Task	RA	PR	EA	BS	VS-U	VS-C	VS-V
Ant jump (5)	0.15	0.81	0.14	0.38	0.85	0.88	0.84
Ant run (5)	0.81	1.05	0.69	0.84	1.06	1.06	1.01
ANYmal walk (5)	1.03	1.03	0.86	1.03	1.02	1.03	1.02
ANYmal jump (5)	0.67	0.62	0.54	0.35	0.70	0.72	0.80

Table 4: Median one-step error of the final model relative to the released model, using reference-policy actions. Evaluation sets are boundary states, independent reference-policy states, and pooled policy-visited states. The last set pools all seventy evaluation policies (Ant only) and is measured on a workstation GPU, with the released model re-measured in the same call. Absolute released-model errors are in Table 5.

Task	Hold-out	RA	PR	EA	BS	VS-U	VS-C	VS-V
Ant jump	Boundary	1.146	0.235	1.339	0.203	0.217	0.224	0.209
	Reference	1.003	0.537	1.236	0.669	0.531	0.483	0.508
	Policy	0.842	0.377	0.801	0.490	0.345	0.382	0.363
Ant run	Boundary	0.316	0.042	0.154	0.017	0.029	0.033	0.031
	Reference	0.737	0.139	0.436	0.161	0.105	0.106	0.117
	Policy	0.860	0.126	0.376	0.293	0.087	0.092	0.089
ANYmal walk (5 runs)	Boundary	1.226	0.795	3.152	0.778	0.755	0.763	0.747
	Reference	1.374	0.801	3.196	0.768	0.787	0.782	0.821
ANYmal jump (5 runs)	Boundary	1.139	0.581	3.298	0.998	0.613	0.596	0.646
	Reference	1.267	0.701	3.346	1.202	0.739	0.713	0.752

Released-model errors and measurement consistency. On Ant jump, released error on the policy-visited set is $1.9\times$ that on the reference-policy set, locating larger errors in the pooled policy-visited distribution (Table 5). Released medians agree within 4% across three GPUs: Ant jump boundary set 1.39–1.42, ANYmal walk reference-policy set 0.157–0.163, and ANYmal jump boundary set 0.166–0.170. Policy-visited set measurements use the workstation GPU with the released model re-evaluated in the same call. Ant values in the table come from the RTX 5090 container, and ANYmal values from the A100 node and container.

Table 5: Released-model median one-step error in MAD_Δ units on the reference-policy action track. Columns correspond to boundary, reference-policy, and pooled policy-visited states. These are the denominators of Table 4.

Task	Boundary	Reference	Policy
Ant jump	1.39	1.27	2.42
Ant run	8.66	2.44	2.67
ANYmal walk	0.156	0.162	—
ANYmal jump	0.167	0.155	—

Return calibration across policies. Exponentiating panel optimism gives a geometric mean over-estimation factor after flooring return ratios below 1 at 1. The released model's Ant run score is only 0.032, corresponding to $1.08\times$, so that task has little optimism to remove. On Ant jump, the strong cross-policy calibration reported in §6.1 includes policies with extreme own-model optimism: all five EA policies are over-estimated by 10^5–$10^9\times$ under their own models, and three RA policies by 10^6–$10^7\times$ (the remaining ratios are 1.2 and 0.8). For BS, three runs give 1.7–$2.8\times$ and two give $263\times$ and $2\times10^5\times$. All candidates come from the study's PPO policy class (Appendix A.4).

Run-to-run stability. Ranges within each condition over five runs on Ant jump: boundary-error ratio BS 0.190–0.216, VS-C 0.138–0.317, EA 1.155–1.503 (above 1 in all five); policy return VS-C

Table 6: Panel optimism: the mean of $\max(0, \log_{10}(J_{\text{model}}/J_{\text{GT}}))$ over 40 policies, averaged over five runs per condition. The released model is shown in the final column.

Task	RA	PR	EA	BS	VS-U	VS-C	VS-V	released
Ant jump	1.351	0.016	0.624	0.116	0.018	0.017	0.017	1.605
Ant run	0.030	0.019	0.031	0.023	0.018	0.018	0.018	0.032

Table 7: Budget-averaged error: mean $\log_{10}$(error/released) over the budget axis from the released model through round 3 (boundary set, reference-policy track; Eq. 2). Each run uses its own final budget for normalization; lower values indicate lower average log error ratios. (a) budget = training transitions, (b) budget = total GT steps including search, exploiter and adversary training. Only runs with all three rounds enter a condition's mean; every row is five complete runs.

Task	Axis	RA	PR	EA	BS	VS-U	VS-C	VS-V
Ant jump	(a)	$+0.05$	-0.45	$+0.07$	-0.57	-0.46	-0.45	-0.46
	(b)	$+0.05$	-0.41	$+0.06$	-0.56	-0.41	-0.41	-0.42
Ant run	(a)	-0.41	-1.12	-0.57	-1.53	-1.24	-1.23	-1.24
	(b)	-0.40	-1.04	-0.51	-1.44	-1.13	-1.12	-1.12
ANYmal walk	(a)	$+0.08$	-0.04	$+0.44$	-0.08	-0.07	-0.06	-0.07
	(b)	$+0.08$	-0.04	$+0.40$	-0.08	-0.06	-0.06	-0.07
ANYmal jump	(a)	$+0.04$	-0.19	$+0.35$	$+0.00$	-0.16	-0.16	-0.15
	(b)	$+0.04$	-0.17	$+0.31$	-0.00	-0.15	-0.15	-0.14

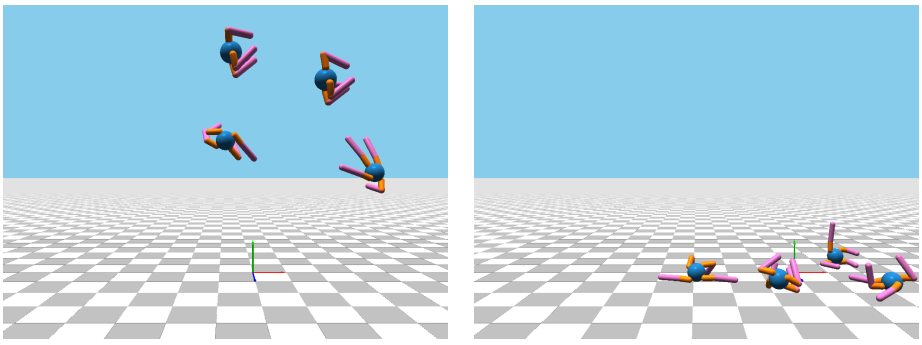


Figure 5: **A policy trained in the** EA**-adapted model, evaluated in the model and GT.** Both views show frame 280 of 300. The four Ant instances remain airborne and inverted in the learned model (left), whereas the same policy produces prone postures in GT (right).

0.79–0.99, EA 0.12–0.16, BS 0.24–0.47; panel optimism BS 0.062–0.268 (the outlier run's model lets its own policy hallucinate $2 \times 10^5\times$). The composition of the pooled policy-visited set is non-uniform by construction (uniform sampling from the pooled visited states favours long-surviving policies: e.g. Ant jump 64 states = RA 8, PR 9, EA 6, BS 6, VS-U 7, VS-C 13, VS-V 15).

Budget-averaged error (descriptive). Using training transitions or total GT steps (including adversary search and BS's probes) gives the same ordering. BS has the lowest mean log boundary-error ratio on each normalized budget axis, and the four strategies using policy-visited states have similar estimates. On jump, RA and EA have nonnegative averages; VS-C−EA is -15.7/-15.8 SD on the two axes (uncorrected $p =$ 1–2 $\times 10^{-4}$; this metric is outside the tested families).

ANYmal results (descriptive). On the two ANYmal tasks, where the initial policy evaluation indicated little room for improvement, prediction-error regression is larger for the error-maximizing adversary: in all twenty (task, hold-out, run) cells EA is worse than released (1.2–8.3×) and so is RA (1.1–1.5×), while every exploiter cell improves (0.5–0.9×); BS leaves ANYmal jump's boundary hold-out unchanged (1.0×) and is 1.2× worse on its on-policy hold-out (the paired-perturbation diagnostic failed its positive control there, so BS and VS-C fail the diagnostic calibration on that task).

ANYmal jump policy outcomes are dominated by a bimodal PPO seed effect—25 of the seventy (condition, run, seed) cells collapse below $0.2\,J_{\mathrm{ref}}$ and 36 exceed 0.9—so we do not read them by condition; all seven strategies sit below released (1.01). Per condition, the collapsed cells (of 10 = 5 runs × 2 seeds) are RA 4, PR 3, EA 3, BS 7, VS-U 3, VS-C 3, VS-V 2; the 36 cells above $0.9\,J_{\mathrm{ref}}$ lie at 0.92–1.14 and the remaining nine non-collapsed cells at 0.20–0.76 (PR 2, EA 4, BS 1, VS-U 1, VS-V 1).

On this task the VS-V run-0 model duplicates the VS-U run-0 model (the exploiter critic's value disagreement had zero variance over that run's candidate pool, so the VS-V weights were constant and its sources coincide with VS-U's uniform sub-sample; Appendix A.4), so there are 68 distinct cells, of which $25/68 = 37\%$ collapse (the two duplicated VS-V cells, 1.10 and 0.56, do not); the five released-model seeds give 0.86, 0.96, 1.01, 1.10, 1.14, i.e. 0 of 5 collapsed. Under a 37% collapse rate, five collapse-free seeds have probability $0.63^5 \approx 0.10$, so the released model's advantage on this task (1.01 vs. 0.62–0.80 for the conditions other than EA and BS) may be a seed effect and is unresolved; more released-model seeds would help assess it. No base-data-only (FT-Only) control was run on ANYmal, so these data do not separate acquisition effects from the fine-tuning schedule as the Ant jump follow-up does (Table 12).

B.2 Collection and contact-event diagnostics

Adversary efficacy and collection states. On Ant jump, released-model mean error in round-1 adversary data is 14.8–19.3 across runs, versus 1.81–1.85 on uniform data (run 0: 19.3 versus 1.84). These collection means are distinct from the hold-out medians. The efficacy check compares the mean one-step error in round-1 EA data with that in RA data; the required ratio exceeds 2. Observed ratios were 6–12 on Ant and 2.5–230 on ANYmal over the five runs; ANYmal walk run 3 passed at 2.5. Collection posture is measured from trunk height and orientation in round-1 data. On Ant run its median height is 0.32 m (10th–90th percentile 0.31–0.66) with 82% of steps below 0.35 m, against the 0.30 m termination threshold and 0.51–0.55 m for RA and PR; on Ant jump 0.43 m (0.36–0.66) with 35% of steps tilted and none inverted, against 0.55–0.59 m and $\leq$ 17% tilted for RA and PR (BS stands at 0.60 m but tilts in 35% of steps; the strategies using policy-visited states VS-U/VS-C/VS-V sit lower, 0.49–0.52 m, and tilt in 33–42% and invert in 9–11% of steps because they fall; Table 11); on ANYmal walk 0.45 m against 0.59–0.65 m. The held-out evaluation sets lie at 0.5–1.6 m (standing, jumping and landing postures).

Coverage of policy-visited states. This exploratory analysis was defined after the main statistical comparisons and before computing coverage; no hypothesis test is attached. To test the reading of §6 that coverage of states visited by model-trained policies drives adaptation, we re-sampled every condition's round-3 collection distribution with the study collector (same start and action rules, from the kept sources, exploiters and adversaries; 40,000 transitions, 4,000 states per (run, condition)) and compared it with the GT states visited by the task's seventy evaluation policies (4,000 states

sub-sampled from 2–2.7 $\times 10^5$). Distances are Euclidean in the re-centred (angular velocity, CoM velocity) representation scaled by the reference's per-dimension MAD; r is the median split-half nearest-neighbour distance of the reference. *Coverage@r* = fraction of reference states within r of a collected state; *fraction near policy samples (r)* = fraction of collected states within r of a reference state (Table 8). Few adversary-collected states lie near the sampled policy trajectories (9% and 0% proximity to policy samples; median distance to the policy samples $2.4r$ and $4.6r$, the largest of all conditions), the four strategies using policy-visited states have the highest, similar sample coverage, and BS covers the *reference* policy's states best (0.54 and 0.30 vs. ≤ 0.50 and 0.00 for the strategies using policy-visited states) while covering the evaluation policies' states no better than uniform data—the data-space form of its boundary/policy-visited error dissociation. Across the seven strategies, and computing each correlation on the column shown in Table 8, coverage and proximity to policy samples rank-correlate with policy return (Spearman 0.71–0.93) and with pooled policy-visited accuracy (0.79–0.96); seven points, descriptive only.

Table 8: Coverage measurement (five-run means). r: Ant jump 7.01, Ant run 3.50 (in MAD units); absolute values depend on r, orderings do not. Ant run coverage is reported at $2r$ because at r every condition covers at most 16% of the reference states (strategies using policy-visited states 7–16%, RA/EA/BS 0%).

Task	Metric	RA	PR	EA	BS	VS-U	VS-C	VS-V
Ant jump	coverage@r	0.312	0.388	0.206	0.311	0.446	0.446	0.446
	fraction near policy samples (r)	0.304	0.393	0.092	0.214	0.396	0.365	0.389
	median NN distance to policy samples (r)	1.15	1.16	2.35	1.24	1.15	1.21	1.14
	coverage of reference-policy states	0.379	0.484	0.282	0.542	0.470	0.466	0.499
Ant run	coverage@$2r$	0.160	0.586	0.000	0.165	0.662	0.640	0.646
	fraction near policy samples (r)	0.000	0.233	0.000	0.000	0.241	0.171	0.196
	median NN distance to policy samples (r)	2.68	1.28	4.56	2.92	1.34	1.61	1.42
	coverage of reference-policy states	0.000	0.004	0.000	0.295	0.000	0.000	0.000

Contact-event divergence in the final models (follow-up; protocol fixed before measuring, no new training). Section 5 established underprediction of contact-event divergence on the released model and on a model fine-tuned on detected failure cases from an earlier study. To connect it to §6 we applied the same instrument to the seven strategies' own round-3 models: at the 31 fixed Ant jump boundary evaluation states, 16 fresh directions per state at $\sigma \in \{10^{-4}, 10^{-3}\}$, rolled 150 steps with zero torque *inside the model under test*, recording divergence when a perturbed rollout's debounced contact-event pattern persistently differs from the unperturbed rollout's (using the same event-comparison rule and debounce constants as the main experiments).

Table 9: Contact-event divergence rate at the fixed Ant jump boundary states, five-run means (range over runs), beside the same conditions' policy return and boundary error. Contact-event sequences diverge in GT in 23.5% of perturbed rollouts.

	Contact-event divergence	Return ($/J_{\mathrm{ref}}$)	Boundary error / released
GT (physics)	0.235	—	—
released model	0.021	0.15	1
RA uniform	0.017 $[0.008, 0.024]$	0.15	1.146
EA Error-MDP	0.022 $[0.004, 0.032]$	0.14	1.339
BS boundary	0.027 $[0.017, 0.037]$	0.38	$\mathbf{0.203}$
VS-C weighted	0.040 $[0.022, 0.064]$	0.88	0.224
PR on-policy	0.042 $[0.026, 0.053]$	0.81	0.235
VS-U uniform-wt	0.043 $[0.025, 0.085]$	0.85	0.217
VS-V value-wt	0.045 $[0.022, 0.066]$	0.84	0.209

The final models retain a substantial contact-response gap: each strategy's five-run mean is 7–19% of the GT divergence rate after three million additional transitions. The visitation strategies nevertheless approximately double the released-model rate, and 17 of their 20 runs exceed the largest RA run. Across the seven strategies, contact-event divergence correlates with policy return (Spearman $+0.71$), while boundary one-step error has correlation -0.50 (lower error should accompany higher return). These are descriptive correlations over seven points, four nearly tied. In particular, BS has the lowest boundary error but a contact-response rate near released and less than half the visitation strategies' return. Both protocols use zero torque, but §5 counts states with any divergent direction; this protocol counts individual perturbation rollouts and uses different states and perturbation sampling. Comparisons are within protocol. Run variability precludes interpreting differences of 0.01 between conditions.

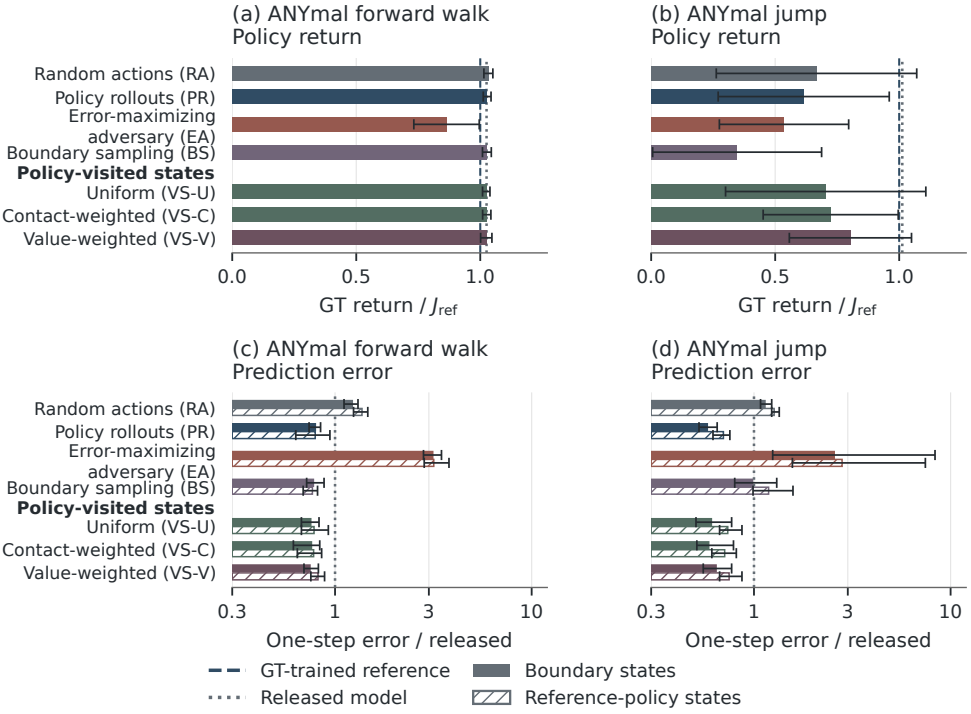


Figure 6: **ANYmal results (descriptive).** Aggregation and graphical conventions follow Fig. 3; prediction error uses logarithmic axes. Both tasks are outside the primary analysis. Policy return includes all ten policy outcomes per condition. Appendix B.1 reports seed-level collapse and the failed positive-control calibration of the jump boundary diagnostic.

B.3 Task-reward and in-simulator adversary follow-ups

The decision rules for these follow-ups were fixed before their measurements. Dann et al. (2026) anticipate that an unconstrained error-maximizer wastes budget on task-irrelevant states and propose a task-constrained variant (their Appendix F.1). Our EA is the unconstrained one, so we ran a fixed-coefficient (Lagrangian-relaxation) approximation of their constrained variant, EA-T, under identical conditions: the same PPO adversary, episode length, reward cap, collection rule, fine-tuning schedule, base set and fixed hold-outs, with the reward changed to $\|(f_{\text{model}} - f_{\text{GT}}) \oslash \text{MAD}_\Delta\|_2 + \alpha\, r_{\text{task}}$. The coefficient α_0 was fixed before any adversary trained, by matching the medians of the two terms over $51{,}200$ steps of the base distribution ($\alpha_0 = 0.069$; medians 0.299 and 4.33). Two further follow-ups, each frozen before its data existed, extend this. *Coefficient sweep* : EA-T at $0.1\,\alpha_0$ and at $10\,\alpha_0$, three runs each, everything else identical, so that the three coefficients span two decades around the median-matched value. *In-simulator adversary* EA-M : the adversary's trajectory evolves *inside the current* EA-M *model* (the context-1 regime of the repository's in-model RL) instead of in the ground truth, so it can enter the model's hallucinated regime as Dann

et al. (2026)'s Algorithm 3 does; its reward is the exact one-step error at the model-visited state, $\|(f_{\text{model}}(s,a) - f_{\text{GT}}(s,a)) \oslash \text{MAD}_\Delta\|_2$ with f_{GT} from a GT-only oracle environment teleported to the model state (the oracle stands in for PASL's learned critic, the same idealisation as in EA), capped at 100; episodes of 100 steps with the task's termination evaluated on the model state and a validity guard (a non-finite state or $|x| > 200$ ends the episode with reward 0); PPO configuration identical to EA (2,048 environments, 300 epochs per round, entropy 0); collection identical to EA (the trained adversary rolled out in GT from resets, $T = 150$). EA-M thus removes exactly EA's confinement to physically reachable states and nothing else. All follow-up conditions: Ant jump, three independent runs per condition, three rounds of $1{,}024{,}000$ GT transitions, the study's byte-identical base set and boundary and reference-policy evaluation sets, policy return from two PPO seeds per run; about 19 GPU-h in total (4.7 for EA-T at α_0, 14 for the nine further runs on an A100 and an H100 node). The same decision table classifies each follow-up by its per-condition three-run mean policy return: at or below $0.20\,J_{\text{ref}}$, between 0.20 and 0.75, or at least 0.75. These ranges retain the pre-specified interpretation categories.

Table 10: Task-reward augmentation (EA-T) and in-model adversary search (EA-M) on Ant jump. Follow-up rows are three-run means with per-run return ranges in brackets; main-experiment rows are five-run means. Boundary and Reference report relative one-step errors. Decision rules use each condition's three-run mean policy return.

	Return ($/J_{\text{ref}}$)	Boundary error	Reference error
released model	0.15	1	1
RA uniform	0.15	1.146	1.003
EA unconstrained error-seeking	0.14	1.339	1.236
EA-T at $0.1\,\alpha_0$ (3 runs)	0.164 [0.123, 0.207]	1.083	1.122
EA-T at α_0 (3 runs)	0.185 [0.164, 0.199]	1.432	1.253
EA-T at $10\,\alpha_0$ (3 runs)	0.148 [0.115, 0.167]	1.135	1.199
EA-M in-simulator adversary (3 runs)	0.158 [0.156, 0.159]	1.373	1.279
BS boundary	0.38	0.203	0.669
Policy-visited PR/VS-U/VS-C/VS-V	0.81–0.88	0.21–0.24	0.48–0.54

All coefficient settings and in-model search fall in the low-return category, with three-run mean return at or below $0.20\,J_{\text{ref}}$ (Table 10). Mean errors exceed those of the released model on both hold-outs. These categories apply to condition means, not individual runs: EA-T at $0.1\,\alpha_0$ has policy return 0.207 in run 2 and boundary error 0.85 in run 1. These descriptive results show little policy improvement under either extension of error-maximizing acquisition.

Large optimism persists in all but one follow-up (condition, run, seed) cell: overestimation factors range from 1.5×10^3 to 5×10^7, with in-model returns of 1.6×10^6–5.5×10^{10} against GT returns of 560–1,520. The remaining cell (EA-T at $0.1\,\alpha_0$, run 0) scores $1{,}421$ in-model and $1{,}469$ in GT. Policy optimization therefore continues to exploit substantial model errors after both extensions of error-maximizing acquisition.

Collection diagnostics. Each round-1 adversary is re-sampled in GT from resets for $40{,}000$ transitions; released-model error is measured on $8{,}192$ sampled transitions. Height is world Y, and posture uses the trunk's local Z axis. Table 11 reports three-run means. Increasing the task-reward coefficient changes collection from crouched to upright postures, while the in-model adversary also collects mostly upright trajectories. Its model rollouts never trigger the validity guard, terminate on 0.02–0.03% of steps, and hit the reward cap on at most 0.8%. Episode error reward decreases over rounds in every run ($1{,}849 \to 913 \to 627$, $1{,}317 \to 884 \to 609$, $1{,}463 \to 638 \to 474$), without comparable policy-return improvement. This batch evaluates policy return and prediction error; optimism on the main study's policy panel was not measured.

B.4 Reference-policy paths, mixtures, a forgetting control, and a constrained adversary

This batch separates collection and update effects through reference-policy data, mixtures, matched-start action changes, and a no-new-data control . All conditions are Ant jump, three independent runs per condition unless marked, three rounds of $1{,}024{,}000$ GT transitions, the study's byte-identical

Table 11: Where the follow-up adversaries collect (Ant jump, round-1 adversary data re-sampled in GT; three-run means for the follow-up conditions). Error = released-model one-step error in MAD_Δ units, mean / median; height = trunk height, median (10th–90th percentile); $< 0.35\,\mathrm{m}$ = fraction of steps below $0.35\,\mathrm{m}$; posture = upright / tilted / inverted fractions; term. = fraction of 150-step trajectories cut by termination. Study rows (EA run 0; strategies using policy-visited states as ranges over PR/VS-U/VS-C/VS-V) from the study's round-1 diagnostic.

	error mean / median	height (m)	$< 0.35\,\mathrm{m}$	upright / tilted / inverted	term.
uniform base set	1.84 / 1.16	0.55 (0.37–0.85)	7.8%	0.98 / 0.02 / 0.00	71%
EA (study, run 0)	19.3 / 21.4	0.43 (0.36–0.66)	4%	0.65 / 0.35 / 0.00	0.1%
EA-T at $0.1\,\alpha_0$	14.0 / 14.5	0.39 (0.36–0.67)	32%	0.83 / 0.17 / 0.00	0.5%
EA-T at $10\,\alpha_0$	17.5 / 3.9	0.58 (0.55–0.72)	0%	0.94 / 0.06 / 0.00	0.3%
EA-M in-simulator adversary	12.0 / 5.3	0.48 (0.44–0.71)	0.1%	0.72 / 0.26 / 0.02	3.2%
Policy-visited PR/VS-U/VS-C/VS-V	2.05–2.28 / 1.24–1.70	0.49–0.59	2–5%	0.47–0.77 / 0.17–0.42 / 0.07–0.11	22–39%

base set, fine-tuning schedule and fixed hold-outs, policy return from two PPO seeds per run. Everything here is descriptive: no statistical comparison in §6 is recomputed and no p-values are attached.

- PR-Ref: acquisition along the frozen *reference* (expert) policy's own path—VS-U with the visitation distribution swapped and nothing else. Candidate pool built with the same pool function as the strategies using policy-visited states so that difficulty is not confounded ; candidates within $6 \times 10^{-3}\,\mathrm{MAD}_v$ of a held-out evaluation set state are dropped, so PR-Ref does not train on the reference-policy set.
- Mix at $\lambda \in \{0.25, 0.5, 0.75\}$: a fraction λ of the equal round budget from the EA adversary, the rest from exploiter rollouts. $\lambda = 0$ is PR and $\lambda = 1$ is EA.
- FT-Only: the same fine-tune schedule with *no* acquisition data at all (base set only)—the forgetting control. Only the base set is used in every round of every run.
- BS-P: BS's boundary start states collected with exploiter-ensemble actions instead of uniform torques. BS was re-run in this batch from the same regenerated CEM boundary source files, so the BS-P – BS contrast is paired.
- EA-C: the Error-MDP adversary with a Lagrangian dual enforcing a mean step-reward threshold, at two floors; EA-CW is the same adversary warm-started from the frozen reference policy, so that it begins inside the feasible set (initialization is the only difference).
- Adversary set: a new held-out evaluation set of adversary-distribution states, and PR, EA, RA, BS re-run as measurement baselines.

Evaluation on the adversary's collection distribution. Before fine-tuning any condition, we trained two independent EA adversaries against the released model and rolled them out in GT from resets. We sampled 64 post-warm-up states, equally from the two runs, and re-centred them as for the reference-policy and policy-visited sets. Their mean step task rewards were 3.489 and 3.507. The adversary set therefore uses the same collection procedure as EA with independent seeds; unlike the boundary set, it does not exclude states near acquisition data. Released-model median one-step error is 4.421 on the adversary set versus 1.270 on the reference-policy set, satisfying the pre-specified ratio criterion of at least 3 with a ratio of 3.48. The boundary error in the same call is 1.394; the zero-action diagnostic gives 3.306, 1.191, and 1.693 on the adversary, reference-policy, and boundary sets.

Repeated baselines. Repeated PR, EA, RA, and BS conditions preserve the main study's ordering (Table 12). Five fresh released-model policies average $0.145\,J_{\mathrm{ref}}$, compared with 0.15 in the main study, and overestimate their own return by 2.7×10^4–2.2×10^{10}. Interpretation thresholds use the pooled between-run SD of the PR, EA, and RA baselines: $s = 0.0564$, giving $\mathrm{SE}_{\mathrm{diff}} = 0.0461$. The low-return threshold is $\bar{J}_{\mathrm{EA}} + 2\,\mathrm{SE}_{\mathrm{diff}} = 0.269$; the high-return criterion requires all three runs to exceed $\bar{J}_{\mathrm{PR}} - 2\,\mathrm{SE}_{\mathrm{diff}} = 0.799$. Even PR misses the latter criterion because its lowest run is 0.796. We therefore report estimates and individual-run values without changing these thresholds.

The controls distinguish five aspects of adaptation. First, FT-Only receives no new data and leaves boundary error at 1.310 times the released model's error, alongside EA at 1.384 and RA at 1.080. Its reference-policy/adversary error ratios of 0.956/0.946 locate the regression primarily at the boundary. This identifies forgetting under the update schedule as a contributor, so the regression cannot be assigned to collection data alone.

Table 12: **Additional acquisition controls on Ant jump.** Return is reference-normalized GT return of fresh PPO policies trained in the final round-3 model, averaged over two seeds per run and then over runs; parentheses give between-run SD. Error columns report median one-step error on boundary, reference-policy, pooled main-experiment policy-visited, and adversary states, relative to the released model re-measured in the same call, using reference-policy actions. GPU-h totals each condition's runs. PR, EA, RA, and BS are repeated baselines.

Condition	Return	per-run return (SD)	Boundary	Reference	Policy	Adversary	runs	GPU-h
PR on-policy exploiter (baseline)	0.891	0.968 / 0.796 / 0.909 (0.088)	0.223	0.534	0.387	0.574	3	6.91
Mix at $\lambda = 0.25$	0.820	0.837 / 0.780 / 0.842 (0.035)	0.339	0.640	0.413	0.528	3	8.30
BS-P boundary + exploiter actions	0.799	0.654 / 0.857 / 0.885 (0.126)	0.151	0.519	0.378	0.522	3	6.91
Mix at $\lambda = 0.5$	0.770	0.784 / 0.852 / 0.675 (0.089)	0.321	0.641	0.467	0.482	3	8.22
Mix at $\lambda = 0.75$	0.697	0.685 / 0.693 / 0.714 (0.015)	0.402	0.722	0.511	0.504	3	8.27
PR-Ref reference-policy path	0.699	0.651 / 0.711 / 0.735 (0.044)	**0.061**	0.200	0.453	0.625	3	3.36
EA-CW constrained, warm start	0.565	0.378 / 0.732 / 0.583 (0.178)	0.118	0.307	0.520	0.609	3	4.68
BS boundary, uniform torques (baseline)	0.390	0.310 / 0.351 / 0.509 (0.105)	0.208	0.616	0.482	0.614	3	3.48
EA-C constrained, cold, floor 13.036	0.213	0.225 / 0.173 / 0.242 (0.036)	1.063	1.232	0.709	0.472	3	4.62
EA-C constrained, cold, floor 7.242	0.187	0.181 / 0.167 / 0.212 (0.023)	1.379	1.181	0.806	0.478	3	4.59
EA Error-MDP adversary (baseline)	0.176	0.163 / 0.196 / 0.170 (0.017)	1.384	1.264	0.797	0.526	3	4.93
FT-Only no acquisition data	0.142	0.141 / 0.139 / 0.147 (0.005)	1.310	0.956	0.943	0.946	3	3.13
RA uniform (baseline)	0.134	0.178 / 0.102 / 0.120 (0.040)	1.080	1.016	0.834	0.725	3	3.42

Second, on matched boundary starts, BS-P reaches $0.799\ J_{\mathrm{ref}}$ against BS's 0.390, demonstrating an action effect conditional on those starts. Its runs span 0.654–0.885 (SD 0.126); run 0 falls 0.145 below the upper decision threshold, and the mean is 0.0002 below that line, leaving the threshold-based label indeterminate. The later policy-visited, uniform-action variant VS-RA (Appendix B.7) evaluates uniform actions from policy-visited starts.

Third, reference-policy collection PR-Ref reaches 0.699 with boundary error 0.061, compared with BS at 0.208 and PR at 0.223, at the lowest cost among this batch's data collection strategies (3.36 GPU-h over three runs, versus 6.91 for PR). Its collected mean step task reward is 11.18–11.55, against the reference policy's 14.484 and EA's 3.46 (3.25–3.52 over nine collection rounds). This extends the evidence for task-relevant collection beyond the model-trained policies.

Fourth, varying the mixture proportion gives 0.891 / 0.820 / 0.770 / 0.697 / 0.176 at adversary fractions 0 / 0.25 / 0.5 / 0.75 / 1. Retaining a quarter of the budget for policy-rollout data achieves 78% of PR's return, or 74% of its improvement above FT-Only at 0.142. Fifth, the cold-started EA family reduces adversary-set error to 0.472–0.526, versus FT-Only at 0.946. The adversary data improve accuracy on their own distribution even as downstream policy return remains low.

Constrained adversaries and initialization. The reward floors are multiples of the reference policy's mean step reward over complete 100-step episodes, measured before collection. Table 13 reports these floors, constraint violations, and collected task reward. Both cold-started variants saturate the dual variable and violate the constraint throughout training; they do not reach the feasible set under the specified optimization protocol. Warm starting improves task reward and downstream return and avoids dual saturation, although constraint violations remain. This comparison includes the reference policy's prior GT training, which was substantially longer than the adversary's few hundred epochs per round.

B.5 Policy selection on full and excluded panels

Model-based policy selection (post-hoc, descriptive). For a model f and candidate set $\mathcal{P}$, predicted return J_f determines the selected policy and its normalized GT-return regret:

$$\widehat{\pi}_f(\mathcal{P}) = \arg\max_{\pi \in \mathcal{P}} J_f(\pi), \qquad \mathrm{Regret}_f(\mathcal{P}) = 1 - \frac{J_{\mathrm{GT}}(\widehat{\pi}_f(\mathcal{P}))}{\max_{\pi \in \mathcal{P}} J_{\mathrm{GT}}(\pi)}. \tag{3}$$

The full panel uses $\mathcal{P} = \Pi$, the forty policies used to measure optimism. Table 14 reports Spearman correlation ρ between model and GT returns, the selected policy's GT rank, and top-1 regret, with percentile-bootstrap intervals over runs. These descriptive analyses use existing evaluations and are separate from the primary and secondary tests.

Optimism and policy selection in the follow-up batch. A fixed panel contains the seed-0 evaluation policy of each condition and run, plus five fresh released-model policies: 44 policies evaluated

Table 13: **Constraint diagnostics for Lagrangian adversaries.** Ant jump, three runs per condition. The reward floor c is relative to the reference policy; λ is the dual variable, capped at 6.91. Violations are the fraction of training epochs below the reward floor. Collected task reward is measured over each condition's nine collection rounds, with ranges in parentheses or separated by a dash; the adversary-set instrumentation runs are excluded. For EA-CW, $\hat{J}$ is mean step task reward over the final 50 training epochs. The first two rows provide reference values.

	floor c	dual λ	epochs violating	collected task reward	Return
reference policy (defines c)	—	—	—	14.484	—
PR-Ref reference-policy path	—	—	—	11.18–11.55	0.699
EA unconstrained	none	—	—	3.46 (3.25–3.52)	0.176
EA-C cold start, $0.5\times$	7.242	cap 6.91, all 9 rounds	100%	3.49	0.187
EA-C cold start, $0.9\times$	13.036	cap 6.91, all 9 rounds	100%	3.53	0.213
EA-CW warm start, $0.9\times$	13.036	0.044–0.633, no saturation	53–66%	≈ 11 ($\hat{J}$ 9.8–13.6)	0.565

Table 14: **Model-based policy selection on the full panel** (post-hoc, descriptive). Each final model scores 40 policies; results average five runs per condition. ρ is the Spearman correlation of model and GT returns. Top-1 regret is the normalized GT-return loss from selecting by model return, with percentile-bootstrap 95% intervals over runs. Rank is the mean GT rank of the selected policy (1 is best). Optimism is defined in Table 6.

	ρ	top-1 regret	true rank of pick	Optimism
Ant jump				
PR	$+0.98$	3.5% [1,5]	3.8	0.016
VS-U	$+0.98$	5.4% [2,9]	4.6	0.018
VS-C	$+0.98$	5.3% [3,7]	5.0	0.017
VS-V	$+0.98$	4.6% [1,8]	4.2	0.017
BS	$+0.81$	73.3% [71,79]	24.4	0.116
RA	-0.15	85.7% [85,86]	32.0	1.351
EA	$+0.05$	89.8% [87,92]	37.8	0.624
released	-0.12	66.6% –	23.0	1.605
Ant run				
PR	$+0.81$	11.8% [4,19]	11.2	0.019
VS-U	$+0.85$	14.1% [6,22]	14.0	0.018
VS-C	$+0.83$	15.0% [7,22]	14.6	0.018
VS-V	$+0.83$	7.8% [2,15]	7.2	0.018
BS	$+0.10$	11.0% [4,20]	10.8	0.023
RA	-0.25	3.5% [4,4]	3.0	0.030
EA	-0.15	36.3% [28,44]	31.4	0.031
released	-0.42	3.5% –	3.0	0.032

in each of 40 models. Panel optimism retains its definition, but the changed panel composition makes the released model in Table 15 the relevant baseline. Task-policy collection reduces optimism substantially, including for PR-Ref and EA-CW, while the no-new-data control retains high optimism. Similar calibration can nevertheless accompany different policy returns: EA-CW and PR-Ref have the same panel optimism but distinct policy returns (Table 12).

Panel membership links selection to policy generation: in 23 of the 40 (condition, run) models, the selected policy was trained under the same condition as the selecting model. The GT-best candidate is PR (run 0, seed 0) ($J_{\mathrm{GT}} = 6{,}328$); thus PR's 0% full-panel regret includes selection of a policy trained under the same condition, whereas BS-P selects this policy from another condition in all three runs. PR-Ref and EA-CW select their own run-0 seed-0 policies, ranked 14th and 22nd of 44, respectively. Their overall rank correlations are close (0.911 and 0.901).

Appendix B.6 examines the effect of excluding policies trained under the same condition.

Table 15: **Panel optimism and policy selection for the additional controls.** Each model evaluates 44 policies over 256 episodes per policy. Optimism follows §3.4; ρ is the Spearman correlation of model and GT returns. Top-1 regret is normalized GT-return loss from selecting by model return, and rank is the selected policy's GT rank (1 is best). Columns are means over three runs, except the single released model; mean ranks need not equal any individual run's rank. Exclusion results are in Appendix B.6.

Condition	Optimism	ρ	top-1 regret	rank of pick
PR on-policy exploiter (baseline)	0.011	+0.982	0.0%	1.0
Mix at $\lambda = 0.25$	0.014	+0.978	5.3%	3.0
BS-P boundary + exploiter actions	0.015	+0.977	0.0%	1.0
Mix at $\lambda = 0.75$	0.019	+0.955	11.1%	5.0
Mix at $\lambda = 0.5$	0.020	+0.974	16.4%	7.0
EA-CW constrained, warm start	0.030	+0.901	60.2%	22.0
PR-Ref reference-policy path	0.030	+0.911	28.5%	14.0
BS boundary, uniform torques (baseline)	0.095	+0.817	72.4%	24.7
EA-C cold, floor 13.036	0.472	+0.378	79.3%	27.7
EA Error-MDP adversary (baseline)	0.770	−0.005	82.4%	32.0
EA-C cold, floor 7.242	0.820	+0.078	82.5%	32.7
RA uniform (baseline)	2.372	−0.552	85.7%	38.0
FT-Only no acquisition data	2.523	−0.527	86.9%	39.0
released model	2.795	−0.581	69.5%	24.0

B.6 Policy selection excluding policies trained under the same condition

Let $a(\pi)$ be the acquisition condition of the model used to train policy π. For a selecting model fitted under condition a, define $\Pi_{-a} = \{\pi \in \Pi : a(\pi) \neq a\}$. We evaluate Eq. 3 with $\mathcal{P} = \Pi_{-a}$, so both selection and regret normalization use only the remaining candidates. Spearman correlation uses the same set. Candidate sets differ across conditions, and the remaining policies still come from the study's PPO procedure; this is an exclusion check rather than an independent common panel.

The original study's strategies using policy-visited states retain high correlations and low regret; BS's regret changes from 73.3% to 7.4%. In the later panel, PR's zero regret becomes 9.4%, while BS-P retains zero. The PR-Ref–EA-CW regret difference narrows from 31.7 to 1.9 percentage points. These descriptive checks do not alter the primary acquisition comparison.

B.7 Uniform actions at policy-visited starts: completed measurements

VS-RA uses the candidate-pool, uniform source selection, teleportation, and perturbation procedure of VS-U, with uniform torques in place of policy actions during collection. Its own model-trained policy ensemble generates the start-state pool each round; the ensemble is not used to choose the subsequent collection actions. The condition shares the base data, three-round schedule, and per-round acquisition budget of the study. The action comparison requires three completed runs of VS-U in the same batch.

Table 16: **Main experiments, Ant jump: exclusion of policies trained under the same condition.** Correlation and regret are means across models of each condition. "Full" uses all panel policies; "excluded" removes policies trained under the same condition as the selecting model before both selection and GT normalization. Original panel size: 40.

Condition	runs	ρ full	ρ excluded	regret full	regret excluded
PR	5	$+0.979$	$+0.980$	3.5%	2.4%
BS	5	$+0.814$	$+0.913$	73.3%	7.4%
RA	5	-0.149	-0.094	85.7%	65.7%
EA	5	$+0.048$	$+0.382$	89.8%	62.2%
released	1	-0.121	-0.305	66.6%	66.6%
VS-V	5	$+0.978$	$+0.979$	4.6%	3.6%
VS-C	5	$+0.977$	$+0.976$	5.3%	4.8%
VS-U	5	$+0.975$	$+0.977$	5.4%	5.4%

Table 17: **Additional controls, Ant jump: exclusion of policies trained under the same condition.** Correlation and regret are means across models of each condition. "Full" uses all panel policies; "excluded" removes policies trained under the same condition as the selecting model before both selection and GT normalization. Original panel size: 44.

Condition	runs	ρ full	ρ excluded	regret full	regret excluded
FT-Only	3	-0.527	-0.449	86.9%	85.6%
PR	3	$+0.982$	$+0.984$	0.0%	9.4%
BS	3	$+0.817$	$+0.871$	72.4%	58.2%
BS-P	3	$+0.977$	$+0.981$	0.0%	0.0%
Mix (0.25)	3	$+0.978$	$+0.983$	5.3%	5.1%
Mix (0.5)	3	$+0.974$	$+0.980$	16.4%	9.2%
Mix (0.75)	3	$+0.955$	$+0.954$	11.1%	11.1%
RA	3	-0.552	-0.715	85.7%	85.7%
EA	3	-0.005	$+0.074$	82.4%	80.5%
EA-C (0.5 floor)	3	$+0.078$	$+0.161$	82.5%	82.2%
EA-C (0.9 floor)	3	$+0.378$	$+0.447$	79.3%	78.4%
EA-CW	3	$+0.901$	$+0.941$	60.2%	20.1%
PR-Ref	3	$+0.911$	$+0.932$	28.5%	18.2%
released	1	-0.581	-0.490	69.5%	69.5%

All three VS-RA runs completed, giving $0.771\,J_{\mathrm{ref}}$ (between-run SD 0.131). Only VS-U run 2 completed ($0.811\,J_{\mathrm{ref}}$); runs 0 and 1 were interrupted after round 2. Consequently no formal conclusion is assigned to the paired comparison. In particular, the single VS-U value is not a three-run comparator and cannot establish equivalence or the primacy of start states. The observed VS-RA performance does show that uniform-torque collection can support substantial recovery when initialized from policy-visited states under this procedure. The earlier BS-P–BS contrast remains an action effect conditional on boundary starts.

Table 18 reports every completed evaluation seed and final policy-track hold-out value. Policy return uses the unchanged reference return $J_{\mathrm{ref}} = 6361.8527$; each prediction error ratio uses the released model measured in the same call.

Table 18: **Completed visited-state action measurements, Ant jump.** Policy return columns are reference-normalized returns; the remaining columns report relative one-step errors on boundary, reference-policy, policy-visited, and adversary states. Blank runs of VS-U are missing, not failures or zero returns.

Condition	run	seed 0	seed 1	mean return	Boundary	Reference	Policy	Adversary
VS-RA	0	1.032	0.789	0.910	0.382	0.646	0.482	0.574
VS-RA	1	0.725	0.578	0.651	0.418	0.712	0.497	0.526
VS-RA	2	0.810	0.695	0.752	0.510	0.708	0.450	0.611
VS-U	2	0.692	0.930	0.811	0.339	0.569	0.399	0.462

C SUPPORTING EXPERIMENTS ON EXPLOITATION AND CONTACT BOUNDARIES

This appendix describes the protocols and results supporting §4 and §5. These experiments use distinct task and evaluation settings and are not additional runs of the main acquisition comparison.

C.1 EXPLOITATION AND PROTOCOL CHECKS

Cross-evaluation and training curves (§4). Ant run, three PPO seeds per condition, 1,000 epochs, 2,048 environments; evaluation over 2,048 GT episodes. 2×2 means: model-trained policy 2638 (in model) / 2454 (GT); GT-trained policy 2004 / 3040. Fixed-policy re-evaluation uses a fresh environment for each policy–simulator combination, records each environment's first episode after a full reset, and resamples episode returns for confidence intervals. For the two training conditions, Δ_A and Δ_B denote model-minus-GT return for model-trained and GT-trained policies, respectively: $\Delta_A = +185.4\ [+178.9, +191.8]$, $\Delta_B = -1019.7\ [-1030.5, -1008.7]$; these are narrow because they are taken over evaluation episodes for fixed policies and do not cover seed-to-seed variation. Training curves (a separate 500-epoch run, checkpoints every 50): in-model return rises to 2572 while GT return plateaus at 2215 (rounded means; gap $+356$ at epoch 500, $+221$ at epoch 200); the GT-trained control of the same run has a -24 gap at epoch 50. Independent re-implementation (new code, different hardware, seeds 0–4) at 500 epochs: 2627.0 / 2154.2 vs. reference 2572 / 2215 ($+2.1\%$ / -2.7%), GT condition 2023.7 / 2980.5 vs. 1999 / 3046; in the same re-implementation, untrained policies showed gaps of $-34.2/+0.2/-56.4$ over three seeds and every condition's epoch-50 gap lay in $[-149, +90]$, all within the pre-specified threshold.

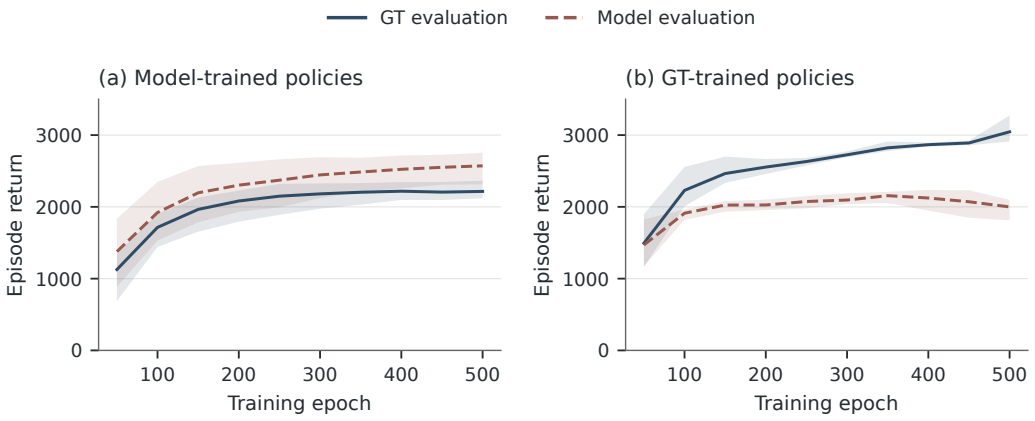


Figure 7: **Training curves on Ant run.** Policies are trained (a) in the released model or (b) in GT and evaluated in both environments. Lines show means across three policy seeds; shaded bands show min–max ranges.

Out-of-distribution jump (§4). Jump reward on the released Ant model; model-trained policies 799 (in model) vs. 101 (GT); optimism $1.08\times$ (run) vs. $7.89\times$ mean and $41\times$ worst seed (jump). Under the fixed-policy re-evaluation protocol just described, mean $6.8\times$ $[6.69, 6.95]$, worst seed $31\times$ $[29.3, 31.8]$. Audit note: an earlier "physical ceiling $\approx$ 238" was a lower bound reached by policies with a corrupted observation and is not used; all ratios are relative to matched baselines.

Context regime. The public training code supervises every position of the ten-step window, including the first position with a single state of context. A full-context check preserves teacher-forced one-step fidelity and increases the frequency and magnitude of optimism, as detailed below. The public rl-games wrapper resets the transformer history every step (effective context 1; verified: 96/96 forwards of length 1). Restoring per-environment context: teacher-forced one-step error dependence on context $\rho = +0.024$ (contact layer, cluster-bootstrap upper bound 0.050), -0.011 on random-torque data; 500-step closed-loop divergence tied; five re-trained policies reached 9.8–

18.4% of the reference in GT—frozen rule "conclusions unchanged". Extreme return overestimation became more frequent with context (1/5 $\rightarrow$ 5/5 seeds, ratios 4.1×10^4–9.1×10^6).

Observation-kernel defect. The public Ant environment writes 29 observation values into a 28-wide buffer; each environment's last value overwrites the next environment's first slot (100% contamination, slots 1–27 intact). Paired bug/fixed comparisons on the standardized jump task showed the defect is not catastrophic (4/4 seeds learn to jump in both; fixed version directionally higher, median $+0.175$ m, 3/4 pairs, $n = 4$). All §6 code uses a corrected 29-dimensional observation.

C.2 Perturbation tests and boundary discovery

Lipschitz nulls and positive control (§5). Paired perturbation tests on Ant run states (compliant GT contact, on-policy states); the statistic is the slope of $\log|\Delta R|$ against $\log\sigma$ across the perturbation ladder, so a Lipschitz response with a modest constant has slope 1 while a slope near 0 means the outcome is scale-free over the probed range. Interpretation: a near-zero slope does not by itself establish a discontinuity ($\tanh(x/\eta)$ is $1/\eta$-Lipschitz yet looks flat to probes coarser than η, and non-smooth is not non-Lipschitz), and the quantity perturbed is the GT return, not the model-smoothed next-state value the bound of Dann et al. (2026) assumes. Three planned null tests: median slope along individual perturbation directions 0.78 at the boundary and in contact and 0.85 in the air, with no boundary-specific departure under these tests. Subsequent checks varied the contact solver, contact stiffness, open- versus closed-loop actions, and evaluation in GT versus the learned model; the following solver comparison was the positive control, while the compliant-contact tests retained the null interpretation. Positive control with XPBD impulsive contact after correcting the velocity convention: fraction of directions with a log-log slope below 0.5: 29.6% $[26.1, 33.4]$ at the boundary against 22.6% $[19.1, 26.0]$ for compliant contact in the same bin, monotone $29.6 > 26.0 > 20.0\%$ (boundary $>$ contact $>$ air); Even pairs with no imposed perturbation ($\delta = 0$) diverged in 18.7% of the XPBD tests. This separate diagnostic records divergence between initially identical batched trajectories and is not evidence of a discontinuity at a measured nonzero perturbation scale. The stiffness-growth endpoint failed its specified criterion; only the paired $\times 10/\times 1$ median growth survived.

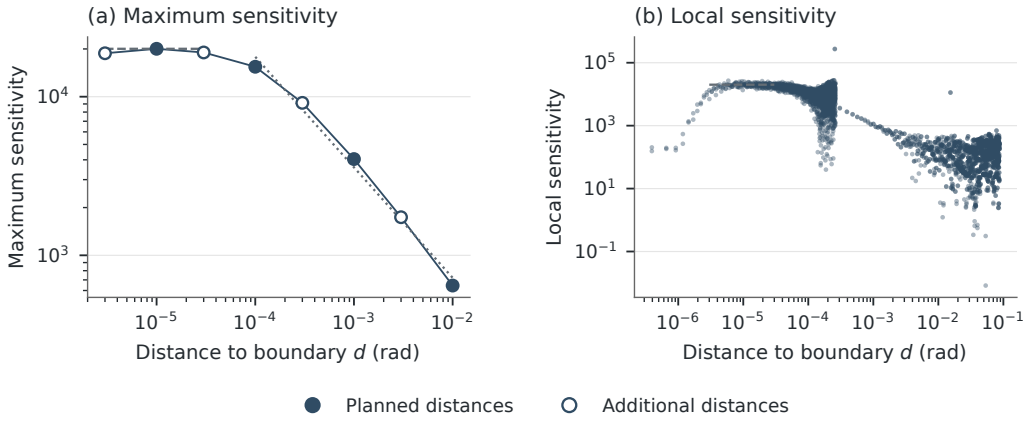


Figure 8: **Sensitivity increases near the cube tip-over boundary and saturates at finite horizon.** Distance d is the initial tilt offset from the boundary, in radians; the rollout horizon is 150 steps. (a) Maximum finite-difference sensitivity across paired perturbations at each d. Filled and open circles distinguish planned and additional distances. The dotted line is the log-log fit over $d \geq 10^{-4}$ (slope -0.70); the dashed segment marks the observed maximum. (b) Local sensitivity, the absolute adjacent-sample change in the same outcome divided by the initial-tilt increment, from the two scan grids. Both panels use logarithmic axes.

Power law and horizon-dependent saturation (§5). Policy-free toppling boundary of a cube resting on one edge in GT (Fig. 8); sensitivity of the tip-over outcome to the initial tilt at controlled distances d from the boundary, maximal sensitivity vs. distance on log-log axes. Fitted slope -0.65

below the ceiling (two estimators, -0.64 and -0.70; a single fit over the whole range is flatter because it averages in the ceiling); the sensitivity ceiling is $\approx 2 \times 10^4$ and rises $\sim 10^3\times$ when the horizon doubles, showing strong horizon dependence of the measured saturation level; cascade mechanism under compliant contact ($16\times$ rougher than the rigid case), smooth under a third setting.

Discovery and verification (§5). An archive groups detections into predefined bins (*archive cells*) to track distinct discoveries; cross-entropy search iteratively concentrates perturbation probes around promising candidates near reference trajectories. Boundary states confirmed on retesting: 85% vs. 0% for random controls on the Ant jump task, and 87.5% vs. 0% in an independent repetition that re-ran discovery from scratch on the jump task (severity correlation 0.60; a third study confirmed on retesting $14/17$ hold-out states, 82.4%); base rate ≈ 19 of $2{,}049$ candidates (0.93%), double-probe labelling $19/24$ vs. $0/48$. RL adversary from resets: pass rate $\approx 1/95$ of random on-policy probes under the matched instrument; 19 archive cells vs. 188 for cross-entropy search in a perturbation shell (a bounded neighbourhood of reference-policy states) with $7\times$ fewer probes. Shell-started controls (same probe budget): a random walk near the detected boundary fills 211 archive cells, 145 of them unvisited by the cross-entropy search, whereas the shell-started RL policy's three-seed union opens only 134 new cells and its pass rate is $0.86\times$ random's (specified threshold $2.0\times$), the three seeds agreeing within ≈ 0.01 percentage points. Cheap selectors (raw score, paired model rollouts, learned ranker): confirmation on retesting 0–5% vs. 85% for physics-in-the-loop search $+$ verification; a prioritizer had a statistically detectable signal that failed the specified enrichment criterion (AUC 0.626, permutation $p = 0.029$; top-336 enrichment $\eta = 0.64$, 90% CI $[0.12, 1.80]$, against the specified requirement that the CI lower bound reach 2.0—the observed upper bound already excludes it).

In a separate model-based selection test, 123 of $7{,}680$ candidates passed initial screening; none of the 20 retested candidates was confirmed in GT.

C.3 Contact-event measurements and earlier adaptation

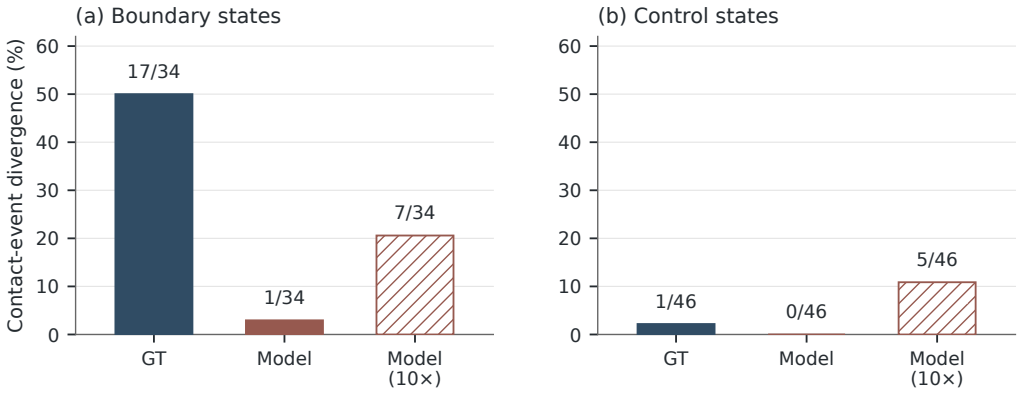


Figure 9: **Contact-event divergence under paired perturbations.** GT and model rollouts use matched initial states and perturbation directions. Panels show (a) 34 verified boundary states and (b) 46 control states. Model ($10\times$) uses tenfold larger perturbations; labels give the number of states with divergent contact-event sequences divided by the number tested.

Paired contact-event measurements and targeted fine-tuning (§5). Figure 9 compares matched GT and model rollouts at verified boundary and control states. Independent re-derivation with new code and new physics re-rolls reproduced $17/34$, $1/34$, $1/46$ and $p = 1.45 \times 10^{-4}$. Fine-tuning on detected failure cases on the same Ant jump task (the 85 states come from the jump fine-tuning loop study, whose seeds are an on-policy harvest of the jump reference policy), four cycles: boundary one-step error $0.722 \to 0.141$ (median; re-measured at the same 85 states in the study measuring contact-event discrepancies after fine-tuning as $0.714 \to 0.142$, the values quoted in the main text)

vs. 0.245 for fine-tuning on random data at equal budget (the 85 states were selected for large initial error, so the five-fold figure includes regression to the mean and the net effect of targeting is the 0.142 vs. 0.245 comparison); contact-event divergence at the same 85 states $4.7\% \to 7.1\%$ vs. GT 51.8% ($+44.7$ pp $[+32.9, +56.5]$, McNemar $p = 4.1 \times 10^{-10}$); targeted vs. random-data fine-tuning $p = 1.000$.

The one-step probe uses zero actions at each boundary state and 16 velocity-perturbed neighbours ($\sigma = 10^{-3}$), with one state of model context. Errors are normalized by the coordinatewise MAD of GT one-step changes in a fixed pre-fine-tuning zero-action calibration batch (floor 10^{-5}). We take the median over the 17 probes per state, then the median over the 85 states. These action, normalization, and aggregation choices differ from the acquisition evaluation in §3.4.

Data-side diagnostic. In two surviving adaptation datasets from the same collector, each $10{,}240$ trajectories $\times$ 100 steps $= 1.02$ M transitions from 256 sources, the perturbed rollouts from every source produce different contact-event sequences (first divergence median 21–22 frames; 94–95% of sources differ even in whether the robot falls), yet among the 2.99 M same-source, same-frame pairs of perturbed rollouts scanned, about $29{,}900$ per dataset have near-identical states (normalized distance ≤ 0.03) and 0 have near-identical state *and* action (minimum action distance 0.242 / 0.159). Because ground truth is deterministic and the torques are drawn independently per environment and frame, the contact-event divergence never appears in the loss as two different outcomes for one input; it appears only as steep variation between nearby inputs, which is consistent with smoothing across a boundary. These data do not isolate an architectural or loss-induced smoothing mechanism; the observed reduction in one-step error and the remaining contact-event discrepancy are separate empirical results.

Earlier adaptation results. Boundary-targeted fine-tuning cut the boundary hold-out tail error (the p95 of that study, not a median; pre-trained 8.70) to 0.57 vs. 1.31 for random data and won on all 17 hold-out states, while Error-MDP data were worse than random there (1.91); its policy-level average endpoint was non-significant ($n = 5$, $p = 0.274$). A repeated jump analysis reproduced the model-side win (medians ratio 1.56 $[1.07, 2.07]$, sign $38/40$) and found the policy side *reversed*: policies trained inside the model fine-tuned on boundary data (the BS counterpart) versus inside the model fine-tuned on random data (the RA counterpart), three seeds each, trained with half of their resets teleported to the verified-boundary shell, evaluated in GT from the same frozen verified-boundary start states (24 perturbations per state, 72 episodes per state), state-paired survival 0.583 vs. 0.761, $\Delta = -0.178$ $[-0.239, -0.115]$—a policy-training outcome measured in physics, not a fidelity statement. An exploiter-in-the-loop measurement-targeted fine-tuning loop did not replicate its real-performance advantage over uniform exploiter adaptation in a five-seed confirmation (2/4), the reason §6 includes the plain policy-rollout strategy PR as a control.

Exploratory cross-robot pilot. Separate from the main Ant/ANYmal comparison, a one-seed pilot found that a pendulum on a raised ground raises the released error $1300\times$ without changing the boundary count ($3 \to 3$), and that on an ANYmal sprint task with no verified boundary all adaptation strategies, and most of all a 1:1 adversary+boundary mixture, worsened the policy (61.5–70.0% of reference vs. 84% for random data).