STATE CORRECTION IN VISUAL CONTROL: CONTACT MECHANISMS AND TASK UTILITY

Anonymous authors
Paper under double-blind review

ABSTRACT

Contact can disrupt the internal state used by a visual controller even when its dynamics parameters are correct. We study how this mismatch affects task performance through controlled interventions in PointMaze, combining a fixed visual world model with calibrated physical priors. Privileged velocity feedback sharply reduces action-adapter trajectory error, whereas the tested increase in endpoint command freedom has little effect. An observation-based repair improves local prediction in development tests, but a separate task evaluation finds little additional goal occupancy over gain scaling. Physical trajectory scoring produces much larger gains, with direct planning achieving slightly higher occupancy at lower cost than reranking neural proposals under the same prior and objective. An independent PointMaze study finds modest gains from visual state resets when the planning model omits contact, while an intact contact prior remains a strong reference. Component interventions distinguish the effects of position and velocity updates. An exploratory analysis of 23 complete PushT scenes shows higher mean object coverage with visual resets and different responses to velocity replacement, extending the analysis to object manipulation. Together, these studies characterize the task value of contact-state correction and show why hybrid visual controllers should be evaluated against direct planning with the same physical information.

1 INTRODUCTION

An agent that learns from experience must turn acquired knowledge into better decisions. This connection motivates the Era of Experience perspective (Silver & Sutton, 2025) and robotic self-modeling, where models inferred through interaction support adaptation to changes in the body (Bongard et al., 2006; Kwiatkowski & Lipson, 2019). Visual world models offer a computational interface for this process: predict candidate futures, compare them with a goal, and execute a promising action sequence (Zhou et al., 2024). When dynamics change, an action adapter can translate commands so that a source predictor remains useful (Hanna & Stone, 2017; Hanna et al., 2021). Yet correctly identifying the change does not ensure that the adapter maintains the state needed to use that knowledge.

Contact makes this distinction concrete (Figure 1). A collision changes a particle's velocity, while an adapter using a free-space update carries momentum that the particle no longer has. The next command is then computed for the wrong motion state, even with correct dynamics coefficients. Giving the command mapping more freedom and correcting its input state are two plausible responses. Comparing these interventions helps distinguish a limitation of action expressiveness from a failure of state propagation.

The practical consequence depends on how the correction enters control. A model selects among candidate trajectories, executes a short prefix, and observes again. Better trajectory predictions need not improve candidate ordering or complete-task behavior, a concern central to task-oriented model evaluation (Farahmand et al., 2017; Lambert et al., 2020). Moreover, a contact prior can supply enough physical structure for a strong direct planner. Its performance is an essential reference when attributing gains to a hybrid visual controller.

We examine these questions through complementary interventions on a fixed visual controller and structured physical planners. In PointMaze, native-dynamics tests isolate the state supplied to an

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.

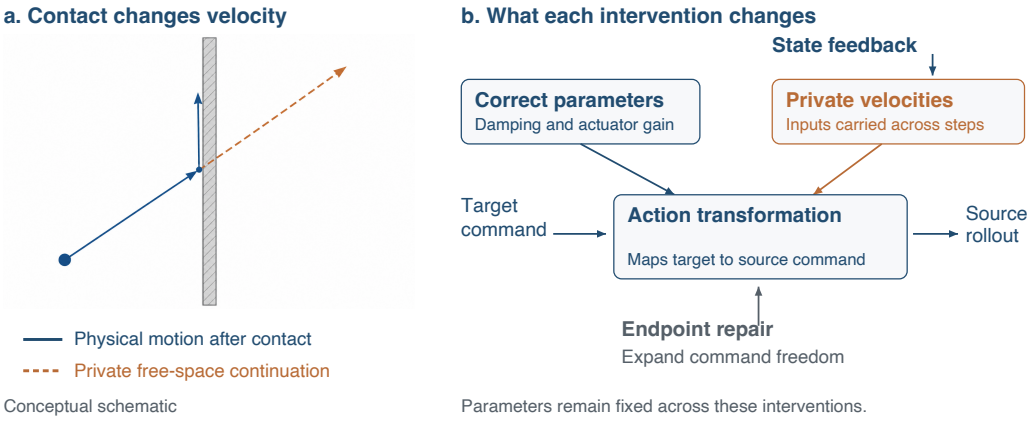

Figure 1: **Contact-induced state inconsistency and diagnostic interventions.** (a) Schematic physical motion after contact (blue) and the adapter's free-space continuation (orange dashed). (b) Velocity feedback supplies simulator velocities to the original action mapping. Endpoint repair jointly adjusts five source commands over the same five-action interval while retaining the free-space endpoint equations.

action adapter; a separate behavioral study measures the incremental value of contact-aware adaptation. We then compare physical scoring of neural proposals with direct planning under the same prior and objective. Finally, observation-based state resets and position/velocity component tests examine when feedback improves physical planning, with an extension to object manipulation in PushT.

The resulting picture is conditional. Privileged velocity feedback sharply reduces the adapter's mechanical error, while an observation-based contact repair adds little mean task benefit over gain scaling. Moving the prior into trajectory planning yields much larger gains, with direct planning providing a strong baseline. New observation-based resets improve PointMaze occupancy under contact omission, while the component results in PointMaze and PushT reveal distinct roles for position updates, velocity estimates, and continued task completion. These comparisons connect a specific computational failure to the decisions through which a repair acquires practical value.

2 Experimental Setting and Study Design

PointMaze requires moving a particle to an image-specified goal in a U-shaped maze. Damping d affects momentum and gain g scales commanded force. The source condition is $(d, g) = (1, 1)$; native diagnostics include damping-only $(4, 1)$ and gain-only $(1, .6)$ shifts. Complete-task tests use the joint shift $(4, .6)$. Scenes vary initial and goal states, support sequences, and search seeds within this environment. PushT extends the state-component analysis to pushing and rotating a T-shaped object (Section 6).

The fixed visual predictor uses a DINOv2 ViT-S/14 encoder architecture (Oquab et al., 2024) and an action-conditioned patch predictor, producing 196×384 features per image. Its input contains three RGB frames and intervening commands; its weights remain fixed throughout evaluation. Policies receive RGB, goal RGB, issued commands, and declared source priors. A shared 32-action reset-support sequence estimates PointMaze damping and gain. Evaluator states and future branches are available only to diagnostic interventions. Architecture and calibration details appear in Appendix A.

Control and search. Each PointMaze action lasts five native .01-second steps. Controllers execute 300 actions, observe and replan every five actions, and search a 25-action horizon with three cross-entropy method (CEM) iterations, 64 candidates and 16 elites. G uses gain scaling and terminal feature distance. E adapts actions using estimated parameters; EC adds a contact prior to state propagation. GP scores G's neural proposals physically; P searches directly with that score. GP

and P interpret the image goal as a location in known geometry. Initial random candidate pools are paired; later adaptive pools and visited states can differ. State-reset experiments use the direct physical planner, without neural prediction calls.

Table 1: Study roles and independent units. Primary contrasts were specified before each prospective study; the PushT row describes the available complete-pair analysis.

Study	Units	Main comparison or analysis
Mechanical diagnosis	120 native cases	Velocity feedback versus endpoint repair; position RMSE
Contact adapter	150 paired scenes	EC versus G; goal occupancy
Task planning	28 paired scenes	GP versus G; goal occupancy
Contact omission	24 paired scenes	Change in the GP–P occupancy gap
Visual state reset	64 paired scenes	Visual reset versus history reconstruction; occupancy
PushT extension	23 complete pairs	State components; mean object coverage

Outcomes and inference. PointMaze occupancy averages goal membership over 300 post-action states, using a radius-.5 goal region. First arrival is 301 for nonarrival and zero for initial inclusion. PushT measures mean object coverage and separately reports reaching and finishing above its native 95% coverage threshold. Scenes are independent units; candidate branches and replans are nested observations. We report paired 95% Student-t intervals, with bootstrap sensitivity analyses in the appendices. Secondary and exploratory intervals are unadjusted. The studies test different interfaces and outcomes, rather than estimating a common mediation effect.

3 CONTACT-INDUCED STATE INCONSISTENCY

Correct coefficients can be applied to the wrong state. Over a held-command interval in free space, one coordinate obeys

$$v^+ = a_d v + b_d g u, \qquad x^+ = x + c_d v + e_d g u, \tag{1}$$

where the coefficients collect the native semi-implicit integration rule. An adapter maintains internal target/source velocities $\hat{v}_t, \hat{v}_s$ and transforms target command u into source command

$$w = \mathrm{clip}_{[-1,1]} \frac{c_t \hat{v}_t + e_t g u - c_s \hat{v}_s}{e_s}. \tag{2}$$

Subscripts t, s denote target and source systems. This mapping matches predicted displacement before saturation; it does not independently match endpoint velocity. Its original internal updates assume free space. Contact can therefore change the native velocity without changing the state used in the next command transformation.

Competing explanations and interventions. The capacity explanation is that a single command cannot match both position and velocity after a damping change. Endpoint repair jointly chooses five bounded source commands over the *same five-action interval* to match the target endpoint. It relaxes command-by-command displacement matching but retains free-space endpoint algebra. The state explanation is tested by current-velocity feedback: simulator velocities of both candidate branches replace the internally propagated values before each command, while the original mapping remains. This diagnostic requires privileged branch information. Disabling wall collision forces supplies a complementary contact intervention while preserving the remaining dynamics.

We first replace the learned predictor by native source mechanics, starting both branches from true state with true parameters. This removes neural and parameter-estimation errors. A one-scene pilot crosses three anchors and 32 fixed sequences with mapping and wall interventions. Under a damping shift, endpoint repair barely changes action-25 position RMSE (.205 to .204), whereas velocity feedback lowers it to .003 and disabling walls lowers the original error to .005. The joint shift follows the same pattern. Branches are nested within the pilot scene. Source and gain-only controls remain exact: admissible $w = gu$ preserves effective inputs, so contact alone does not preclude exact transport.

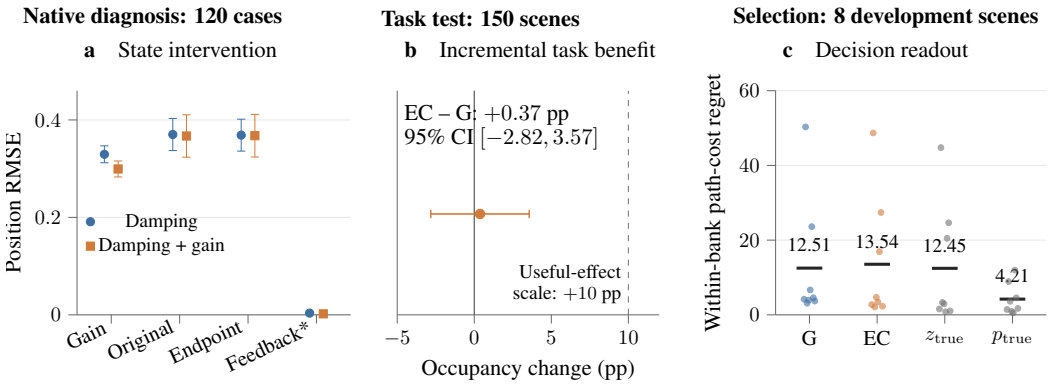

Figure 2: **State correction, task benefit, and decision readout.** (a) Native source/target dynamics with true parameters: mean action-25 position RMSE and marginal 95% intervals over 120 cases. Endpoint repair chooses five commands over the same interval; feedback (*) uses evaluator-only branch velocities. (b) EC versus G occupancy on 150 independent scenes, with paired 95% interval and the prespecified useful-effect scale. (c) A separate eight-scene development diagnostic on shared candidate banks: scene-mean path-cost regret and overall means. Evaluator-only z_{true} and p_{true} select by realized terminal feature and position distance. Regret is excess summed squared goal distance over 25 actions relative to the best candidate in the bank.

Independent confirmation of the state intervention. After eight development scenes, a fixed test uses 120 fresh test cases, each containing 16 clipped Gaussian 50-action sequences under all four parameter conditions and both wall settings. The primary comparison averages damping-only/joint feedback-minus-endpoint-repair coordinate-position RMSE at action 25 with walls. The prespecified .05-unit mechanical margin is half the particle radius. The mean difference is $-.365$, with paired 95% interval $[-.403, -.328]$ (Figure 2). The secondary endpoint-repair-minus-original difference is $-.0003$ $[-.0036, .0030]$.

Correcting the state supplied to the tested mapping reduces error much more than extra endpoint freedom. Appendix A gives the corresponding error decomposition. A deployable correction must estimate branch states from observations and a contact model.

4 Incremental Utility of Contact-Aware Adaptation

Observation-based state correction. EC replaces the adapter's free-space propagation with a specified source contact prior: known geometry, contact law, mass, gear, and integration rules. Camera translation comes from visible walls, scale from source RGB-command calibration, and velocity from observed displacements and commands. The prior then propagates these estimates along candidate action sequences. On development anchors, this estimated-parameter model reduces position RMSE from .238 to .046 under the damping shift and from .298 to .082 under the joint shift.

The independent control study uses 150 paired scenes and one fixed visual predictor. It compares the unadapted controller F, the free-space adapter E, gain scaling G, and contact-aware adaptation EC. Two diagnostic variants, O and OC, give the corresponding adapters true parameters. The prespecified primary is EC–G goal occupancy, testing the incremental value of contact reasoning beyond gain scaling, with a useful-effect scale of 10 percentage points (pp). Eight development scenes informed the fixed sample size and a 5-pp interval-half-width precision target.

Incremental benefit beyond gain scaling. EC achieves 44.88% occupancy against G's 44.51%. Their paired mean difference is $+.37$ pp, with 95% interval $[-2.82, 3.57]$ (Figure 2b). The upper confidence limit is below the prespecified 10-pp useful-effect threshold. G's occupancy leaves substantial room for improvement, and EC changes commands relative to both G and E in all 150 scenes. Reset estimates are accurate, with damping/gain mean absolute errors .0044 and 3.95×10^{-6}.

EC also incurs higher planning cost and fewer goal arrivals than G: 295.33 versus 216.79 seconds and 117/150 versus 126/150 arrivals. The secondary arrival difference is -6 pp $[-10.27, -1.73]$, and the squared-distance sum increases by 65.78 [11.15,120.41]. Occupancy improves against F and E by +15.39 and +3.69 pp, respectively. The analysis includes all 900 episodes, including 171 nonarrivals and 16 initially-inside scenes; full comparisons appear in Appendix C.

Why a prediction gain need not improve selection. For a realized terminal feature z, prediction $\widehat{z}$, goal feature z_g, and $e = \widehat{z} - z$, let $L = D^{-1}\|z - z_g\|^2$ over D feature coordinates. Then

$$\widehat{L} - L = \frac{\|e\|^2}{D} + \frac{2\langle e, z - z_g\rangle}{D}. \tag{3}$$

Prediction MSE controls only the first term. The signed, goal-dependent cross term can change differently across candidates, so lower mean MSE does not guarantee better score ordering. Even exact feature scores need not rank physical task outcomes correctly: terminal resemblance differs from reaching and occupying a goal over time.

The eight-scene development diagnostic examines these distinctions on common candidate banks from matched visited histories. EC lowers feature MSE relative to G by .0101, but its path-cost regret difference is $+1.03$ $[-6.56, 8.61]$. Selecting with *realized* terminal features gives mean regret 12.45, close to G's 12.51; selecting by realized terminal physical distance gives 4.21 (Figure 2c). Even with prediction error removed, the terminal feature criterion selects candidates with substantial path-cost regret. This motivates evaluating the physical trajectory and its intermediate states directly.

5 Task-level planning with contact priors

Physical trajectory scoring. GP and P reuse EC's calibrated target contact prior. For candidate u, it predicts positions $\widehat{p}_{t+k}(u)$; the visible goal supplies $\widehat{p}_g$. Both planners use

$$\widehat{C}(u) = \frac{1}{25}\sum_{k=1}^{25} \rho(\widehat{p}_{t+k}(u), \widehat{p}_g)^2, \tag{4}$$

where ρ is a source-map path-distance surrogate around the inner wall, conservatively expanded by particle radius. Small predicted penetrations are projected to a wall face and penalized. The score uses known geometry and interprets the goal as a physical location. Averaging over the path penalizes prolonged distance even when a candidate ends near the goal. In free space, this reduces to a finite-horizon quadratic position cost, which depends on intermediate states as well as the endpoint (Appendix G). Occupancy remains the evaluation metric.

GP collects all 192 candidates from G's three neural CEM iterations and selects the minimum physical score. P runs CEM directly on that score, without neural-predictor calls (Table 2). P thus provides a direct baseline under the same physical information and task objective. The planners differ in how they generate candidates: G's feature scores guide GP's proposal search, while physical trajectory costs guide P's search.

Table 2: Runtime computation in the intact-prior comparison. All methods receive RGB, goal RGB, issued commands, and reset support; none receives evaluator states or future labels. G uses gain scaling, EC repairs its action interface, GP scores neural proposals physically, and P searches directly.

Method	Contact prior	Physical goal score	Neural prediction
G	No	No	Yes
EC	Yes, action interface	No	Yes
GP	Yes, task trajectories	Yes	Yes
P	Yes, task trajectories	Yes	No

Task performance and planning cost. The confirmation uses 28 fresh paired scenes and four controllers, giving 112 complete episodes. GP improves occupancy over G by $+53.45$ pp [43.52,63.39],

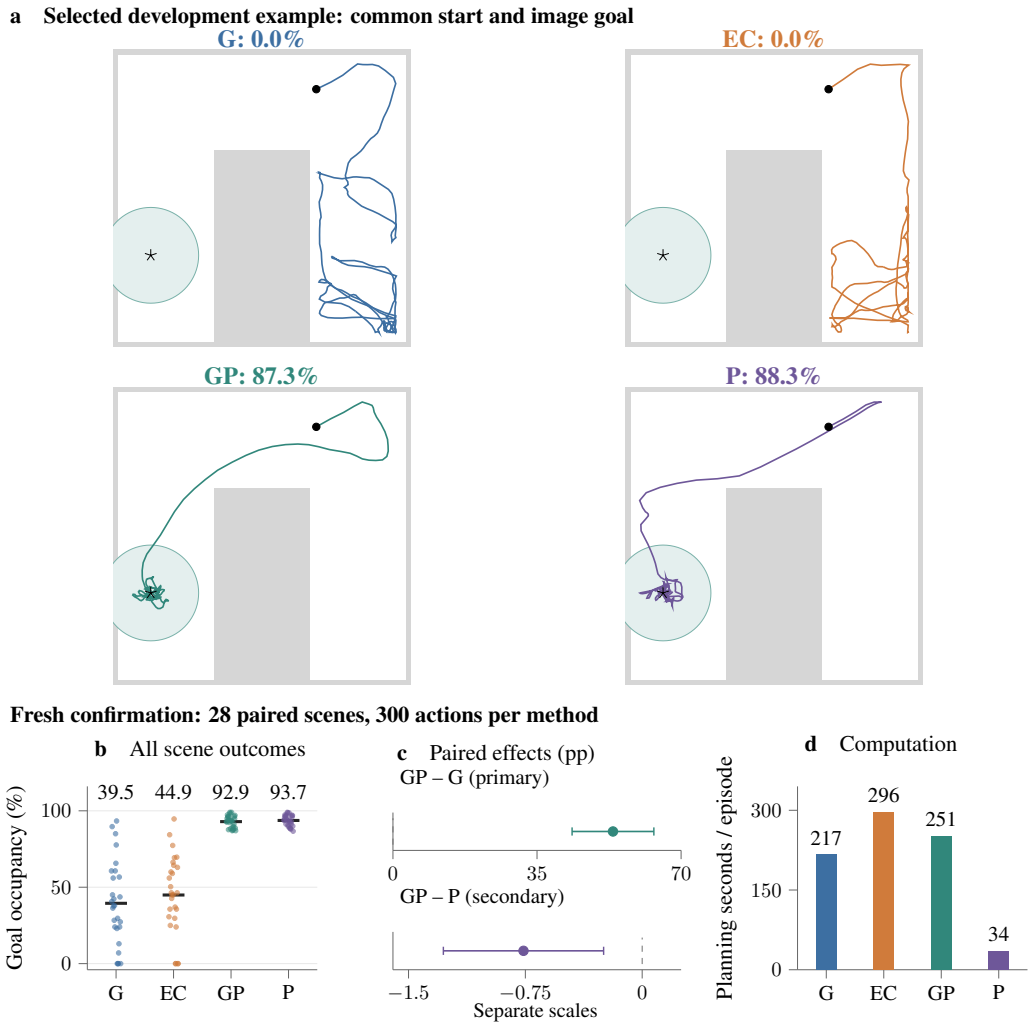

Figure 3: **Task-level planning improves occupancy; direct planning achieves slightly higher occupancy at lower cost.** G uses gain-scaled neural planning; EC adds contact-aware action adaptation. GP scores G's proposals using calibrated contact dynamics and known geometry; P directly optimizes the same physical score. (a) A development scene selected before confirmation to illustrate scoring mismatch. G and EC do not arrive, whereas GP and P go around the wall. Black circles mark the common start, stars the goal, and shaded disks its radius-.5 region; labels give 300-action occupancy. (b–d) Independent confirmation on 28 fresh paired scenes. Dots in (b) are scene occupancies; bars and numbers are means. (c) Paired 95% intervals show the prespecified GP–G effect and secondary GP–P contrast on different scales. (d) Mean planning time excludes shared initialization, support, and rendering; all methods use the same Torch/CUDA framework.

exceeding the prespecified 10-pp useful-effect scale for this primary contrast. Every scene's difference is positive; 1,654 of 1,680 executed first chunks differ from the neural selection. Three initially-inside scenes and all valid nonarrivals are included.

P achieves 93.69% occupancy versus GP's 92.93%, at 34.41 versus 250.64 planning seconds (Figure 3). The secondary GP–P difference is $-.76$ pp $[-1.28, -.25]$; P uses 86.3% less planning time. Both arrive and finish inside the goal in all 28 scenes. These timings measure the common Torch/CUDA implementation. High occupancy is therefore attainable with direct physical planning under the supplied prior and objective. Full per-method results appear in Appendix D.

Objective components. GP changes physical prediction, goal readout, and time aggregation together. An exploratory component comparison retains the same prior and search on eight development scenes. Integrated Euclidean cost exceeds terminal Euclidean cost by +27.25 pp [17.03,37.47]; adding the navigation geometry yields +.17 pp $[-.82, 1.15]$. All variants arrive, but the terminal score finishes inside the goal in only five of eight scenes. Temporal aggregation has the larger observed effect in this development comparison; the fresh-scene GP–G result evaluates the combined intervention. A local-feature weighting that did not improve performance is reported in Appendix E. The secondary EC–G occupancy interval is $[-1.84, 12.70]$ in the smaller 28-scene cohort.

6 Observation-based state resets and task components

The preceding comparisons show that the location of a physical prior matters. We next separate its role in reconstructing the current state from its role in predicting future contact, and ask whether recent observations can replace an inaccurate reconstruction.

From contact omission to visual feedback. An initial 24-scene study removes private wall forces during history reconstruction and future prediction while retaining real walls and the goal map. Its primary interaction, the change in the GP–P occupancy gap, is +4.01 pp $[-3.27, 11.30]$; the relative benefit of neural proposals remains uncertain. A secondary visual-reset comparison motivates the independent 64-scene study below. Full initial-study results appear in Appendix K.

Position and velocity interventions. Let q_t be position extracted from calibrated RGB. The visual estimate uses

$$v_t^V = \mathrm{clip}_{[-5,5]} \frac{q_t - q_{t-\Delta}}{\Delta}, \qquad p_t^V = q_t + \ell v_t^V, \tag{5}$$

with $\Delta = .25$ seconds and calibrated image delay ℓ. The history estimate H propagates earlier observations and up to ten commands with private wall forces omitted. Four combinations, HH, VH, HV, VV, select position and velocity from history (H) or recent observations (V). Each initializes the same canonical simulator before future planning; $V0$ retains visual position but sets initialized velocity to zero. Future contact is either retained (C) or omitted (F). Legacy P_{FF} and P_{CC} retain the original history-replay implementation. The shared score and search settings are those of direct planning in Section 5. Unlike the privileged feedback in Section 3, this update changes only the state at a replanning boundary.

For task outcome J, the position effect is $\frac{1}{2}[(J_{VH} - J_{HH}) + (J_{VV} - J_{HV})]$, averaging replacement with history and visual velocity. The velocity effect averages $(J_{HV} - J_{HH})$ and $(J_{VV} - J_{VH})$ in the same way.

PointMaze: independent task gains and component effects. Across 64 fresh paired scenes and 768 complete episodes, visual reset increases occupancy from 89.71% to 92.20%: +2.48 pp [0.29,4.68] (Figure 4a). The median difference is +.67 pp; 43 scenes improve, ten worsen, and eleven are unchanged. The canonical HH reset alone has zero occupancy difference from P_{FF} in every scene, separating coordinate replacement from reset implementation. Position replacement has effects of +1.42 pp [0.31,2.53] with future F and +2.65 pp [1.66,3.64] with future C. Corresponding velocity effects are +1.07 pp $[-.12, 2.25]$ and $-.44$ pp $[-1.32, .43]$. These component contrasts describe closed-loop controller changes, whose later histories can differ.

Discarding velocity has a much larger effect than replacing its estimate: $V0$ loses 18.73 pp [13.84,23.63] with future F and 28.09 pp [23.18,33.00] with future C. Both variants reach the goal in all scenes, so arrival alone conceals the loss of continued occupancy. Against the intact-prior baseline, $R_{VV,C} - P_{CC}$ is $-.35$ pp $[-.75, .04]$. Thus the observed benefit under contact omission coexists with a strong intact-prior reference.

PushT: component effects in object manipulation. We apply the same component design to a native PushT variant (Chi et al., 2023). Known colors, camera geometry and the T outline provide visual position and orientation; finite differences over .5 seconds estimate linear and angular velocity. The history intervention omits pusher–object contact, while every future prediction retains it. The physical prior uses known stationary dynamics. CEM scores terminal coverage, object pose error, and pusher–object distance; controllers again execute 300 actions. Six methods are compared on 23

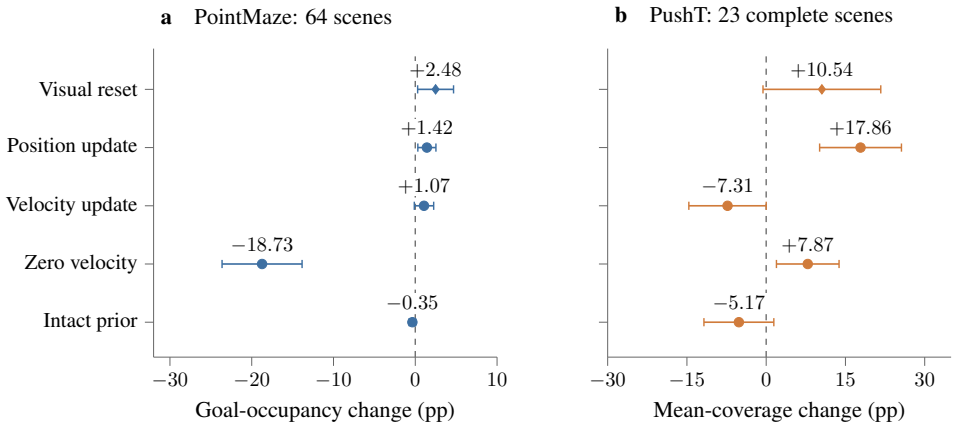

Figure 4: **Visual state resets and their components.** Points show paired mean differences with 95% intervals. (a) PointMaze uses 64 fresh scenes and goal occupancy; the visual-reset primary compares $R_{VV,F}$ with P_{FF}. Position/velocity effects average the two counterpart-controlled contrasts with future F; zero velocity compares $V0$ with VV. The intact-prior row compares $R_{VV,C}$ with P_{CC}. (b) Exploratory PushT results use 23 complete pairs and mean object coverage: corresponding contrasts use HH_F history, all future contacts, and intact-history reference H_C. Axes use task-specific outcomes and scales.

complete paired scenes. This exploratory complete-case sample comes from 24 attempted scenes: one history-reconstruction episode loses the object from view after 255 actions, leaving its full-horizon outcome unavailable. The original full-sample primary and the complete-case analysis are distinguished in Appendix R.

Visual reset raises mean coverage from 51.41% to 61.95%, a paired difference of +10.54 pp $[-.60, 21.69]$ (Figure 4b). Position replacement has a +17.86-pp effect [10.11,25.60], whereas velocity replacement gives -7.31 pp $[-14.64, .01]$. Setting velocity to zero increases coverage by +7.87 pp [1.94,13.79]. The intact-history controller achieves 67.12% coverage; the $VV - H_C$ difference is -5.17 pp $[-11.78, 1.45]$. The visual-reset interval spans zero, while the component results associate the clearest coverage increase with position replacement.

Table 3: PushT outcomes on the same 23 complete paired scenes. Coverage is averaged over 300 actions; arrival and final success use the native 95% coverage threshold. H_C retains contact during history reconstruction; HH_F omits it. VH_F/HV_F replace only position/velocity, VV replaces both, and $V0$ sets initialized velocity to zero.

Method	Mean coverage (%)	Arrival	Final success
H_C	67.12	7/23	6/23
HH_F	51.41	4/23	4/23
VH_F	70.06	0/23	0/23
HV_F	44.89	4/23	3/23
VV	61.95	7/23	2/23
$V0$	69.82	5/23	3/23

Coverage and completion distinguish two consequences of the update. Position-only replacement produces the highest mean coverage but never reaches the 95% threshold, while intact history more often finishes above it (Table 3). The opposite responses to zero velocity in PointMaze and PushT also show why position correction and velocity initialization warrant separate tests. A finite difference averages motion across an observation interval; contact and rotation can make this average differ from the velocity required for the next prediction.

7 Related work

Experience, self-modeling, and action grounding. Robotic self-modeling links interaction-derived models to adaptation (Bongard et al., 2006; Kwiatkowski & Lipson, 2019). Grounded action transformation adjusts interfaces across systems (Hanna & Stone, 2017; Hanna et al., 2021); GARAT learns transformations from observation (Desai et al., 2020). GAT already recognizes contact-related limits to action equivalence. Our native interventions distinguish state propagation from endpoint command freedom within a specific analytic adapter.

State estimation and predictive control. World models support learning in imagined environments (Ha & Schmidhuber, 2018); PETS and MBPO study uncertainty and model use (Chua et al., 2018; Janner et al., 2019). Visual Foresight and PlaNet plan from visual observations (Ebert et al., 2018; Hafner et al., 2019), while DINO-WM predicts pretrained patch features (Zhou et al., 2024). State estimation followed by planning is standard MPC (Rawlings et al., 2024). Contact-UAN further shows that contact-inconsistent initialization can cause an actuator residual to absorb state-reset errors, and estimates contact-consistent states from proprioception before residual learning (Fey et al., 2026). We examine this distinction in online RGB control through component interventions, complete-task outcomes, and direct physical-planning comparisons.

Prediction objectives and decision value. Value-aware learning and value equivalence connect model quality to downstream computations (Farahmand et al., 2017; Grimm et al., 2020); predictive fit can diverge from control performance (Lambert et al., 2020). Recent work examines trajectory reachability (Li et al., 2026), decision-metric alignment (Wang et al., 2026), and update/hold utility (Li & Kim, 2026). GPC uses predictive world models to rank and optimize generative policy proposals (Qi et al., 2026). Our comparison asks what performance is supported by the supplied physical prior itself, alongside its use in a hybrid controller.

8 Discussion and conclusion

Accurate parameters specify how a state evolves, but do not correct a state already made inconsistent by contact. The native interventions locate this failure in the tested adapter. Its observation-based repair improves local prediction, yet adds little occupancy beyond gain scaling. Physical trajectory scoring produces a much larger task gain; direct planning with the same prior and objective is slightly better and substantially cheaper than neural proposal reranking. Since candidate distributions differ, this is a comparison of complete planners under common physical information.

The state-reset studies clarify where observation feedback contributes. In PointMaze, recent-image resets improve planning with omitted contacts, while the intact prior remains competitive. Position and velocity interventions have distinct effects, and PushT separates improved coverage from sustained task completion. The finite-difference update is a simple feedback baseline; its value here is to test the practical implications of the diagnosis. The planning-time reduction in the reset implementation largely comes from computing history propagation once per replan rather than per candidate, as the coordinate-preserving reset comparison also shows.

These results suggest evaluating adaptation at three connected levels: the state supplied to the computation, the quality of decisions on shared histories, and complete-task performance against baselines receiving the same added information. Our evidence covers one fixed visual predictor, a specified PointMaze shift, and structured RGB extraction with known mechanics. PushT adds an object-manipulation setting for the component analysis. The tests do not include natural domain drift or real-time hardware control; simulation pauses during planning. Observation-based resets inside the original neural action interface remain a distinct empirical question because the new reset studies use physical scoring. Across these settings, the central design lesson is to evaluate both where physical knowledge enters a controller and the task performance that knowledge supports on its own.

AI USE STATEMENT

AI assistance was used in research design and critique, code development and experiment supervision, literature search and citation checking, analysis of experimental results, figure preparation, and manuscript drafting and editing. Reported measurements come from executed simulator and model computations. The conceptual contact illustration in Figure 1(a) was generated with AI assistance and combined with programmatic labels; it is not an experimental trajectory. Quantitative plots and the trajectories in Figure 3 use recorded computational results. The authors are responsible for the scientific claims, references, and disclosure.

REPRODUCIBILITY STATEMENT

The appendices specify the model architecture, observations and priors available to each method, action and planning schedules, evaluation units, statistical procedures, and complete method comparisons. They also report development alternatives, negative results, and limitations of the diagnostic interventions. The accompanying reproducibility documentation preserves configurations, model identifiers, analysis scripts, and experiment records needed to trace the reported results.

REFERENCES

Josh Bongard, Victor Zykov, and Hod Lipson. Resilient machines through continuous self-modeling. *Science*, 314(5802):1118–1121, 2006. doi: 10.1126/science.1133687. URL https://www.science.org/doi/10.1126/science.1133687.

Cheng Chi, Siyuan Feng, Yilun Du, Zhenjia Xu, Eric Cousineau, Benjamin CM Burchfiel, and Shuran Song. Diffusion policy: Visuomotor policy learning via action diffusion. In *Proceedings of Robotics: Science and Systems*, 2023. doi: 10.15607/RSS.2023.XIX.026. URL https://www.roboticsproceedings.org/rss19/p026.html.

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*, volume 31, 2018. URL https://papers.nips.cc/paper/2018/hash/3de568f8597b94bda53149c7d7f5958c-Abstract.html.

Siddharth Desai, Ishan Durugkar, Haresh Karnan, Garrett Warnell, Josiah P. Hanna, and Peter Stone. An imitation from observation approach to transfer learning with dynamics mismatch. In *Advances in Neural Information Processing Systems*, volume 33, 2020. URL https://arxiv.org/abs/2008.01594.

Frederik Ebert, Chelsea Finn, Sudeep Dasari, Annie Xie, Alex Lee, and Sergey Levine. Visual foresight: Model-based deep reinforcement learning for vision-based robotic control. *arXiv preprint arXiv:1812.00568*, 2018. URL https://arxiv.org/abs/1812.00568v1. Version 1, 3 December 2018.

Amir-Massoud Farahmand, Andre Barreto, and Daniel Nikovski. Value-aware loss function for model-based reinforcement learning. In *Proceedings of the 20th International Conference on Artificial Intelligence and Statistics*, volume 54 of *Proceedings of Machine Learning Research*, pp. 1486–1494, 2017. URL https://proceedings.mlr.press/v54/farahmand17a.html.

Nolan Fey, Khai Nguyen, Yash Prabhu, Gabriel B. Margolis, and Pulkit Agrawal. Contact-uan: Unsupervised actuator networks as a residual world model for humanoid control. Author-hosted manuscript, 2026. URL https://contact-uan.csail.mit.edu/static/contact-uan.pdf. Accessed 26 September 2026.

Christopher Grimm, Andre Barreto, Satinder P. Singh, and David Silver. The value equivalence principle for model-based reinforcement learning. In *Advances in Neural Information Processing Systems*, volume 33, 2020. URL https://proceedings.neurips.cc/paper/2020/hash/3bb585ea00014b0e3ebe4c6dd165a358-Abstract.html.

David Ha and Jürgen Schmidhuber. Recurrent world models facilitate policy evolution. In *Advances in Neural Information Processing Systems*, volume 31, 2018. URL https://papers.nips.cc/paper/2018/hash/2de5d16682c3c35007e4e92982f1a2ba-Abstract.html.

Danijar Hafner, Timothy Lillicrap, Ian Fischer, Ruben Villegas, David Ha, Honglak Lee, and James Davidson. Learning latent dynamics for planning from pixels. In *Proceedings of the 36th International Conference on Machine Learning*, volume 97, pp. 2555–2565, 2019. URL https://proceedings.mlr.press/v97/hafner19a.html.

Josiah P. Hanna and Peter Stone. Grounded action transformation for robot learning in simulation. In *Proceedings of the AAAI Conference on Artificial Intelligence*, 2017. URL https://www.cs.utexas.edu/~jphanna/papers/hanna2017grounded.pdf.

Josiah P. Hanna, Siddharth Desai, Haresh Karnan, Garrett Warnell, and Peter Stone. Grounded action transformation for sim-to-real reinforcement learning. *Machine Learning*, 110:2469–2499, 2021. doi: 10.1007/s10994-021-05982-z. URL https://link.springer.com/article/10.1007/s10994-021-05982-z.

Michael Janner, Justin Fu, Marvin Zhang, and Sergey Levine. When to trust your model: Model-based policy optimization. In *Advances in Neural Information Processing Systems*, volume 32, 2019. URL https://proceedings.neurips.cc/paper/2019/hash/5faf461eff3099671ad63c6f3f094f7f-Abstract.html.

Robert Kwiatkowski and Hod Lipson. Task-agnostic self-modeling machines. *Science Robotics*, 4(26):eaau9354, 2019. doi: 10.1126/scirobotics.aau9354. URL https://www.science.org/doi/10.1126/scirobotics.aau9354.

Nathan Lambert, Brandon Amos, Omry Yadan, and Roberto Calandra. Objective mismatch in model-based reinforcement learning. In *Proceedings of the 2nd Conference on Learning for Dynamics and Control*, volume 120 of *Proceedings of Machine Learning Research*, pp. 761–770, 2020. URL https://proceedings.mlr.press/v120/lambert20a.html.

Anqi Peter Li and Kaden Kim. Measuring the value of world-model updates: A counterfactual utility protocol for continual adaptation. *arXiv preprint arXiv:2609.10954*, 2026. URL https://arxiv.org/abs/2609.10954v1. Version 1, 10 September 2026.

Liangyu Li, Shengzhi Wang, Libin Qiu, Mingliang Xiong, and Qingwen Liu. World model control by trajectory reachability metrics. *arXiv preprint arXiv:2605.22164*, 2026. URL https://arxiv.org/abs/2605.22164v2. Version 2, 31 August 2026.

Maxime Oquab, Timothée Darcet, Théo Moutakanni, Huy V. Vo, Marc Szafraniec, Vasil Khalidov, Pierre Fernandez, Daniel Haziza, Francisco Massa, Alaaeldin El-Nouby, Mahmoud Assran, Nicolas Ballas, Wojciech Galuba, Russell Howes, Po-Yao Huang, Shang-Wen Li, Ishan Misra, Michael Rabbat, Vasu Sharma, Gabriel Synnaeve, Hu Xu, Hervé Jegou, Julien Mairal, Patrick Labatut, Armand Joulin, and Piotr Bojanowski. DINOv2: Learning robust visual features without supervision. *Transactions on Machine Learning Research*, 2024. URL https://openreview.net/forum?id=a68SUt6zFt.

Han Qi, Haocheng Yin, Aris Zhu, Yilun Du, and Heng Yang. Inference-time enhancement of generative robot policies via predictive world modeling. *arXiv preprint arXiv:2502.00622*, 2026. URL https://arxiv.org/abs/2502.00622v4. Version 4, 11 March 2026.

James B. Rawlings, David Q. Mayne, and Moritz M. Diehl. *Model Predictive Control: Theory, Computation, and Design*. Nob Hill Publishing, 2 edition, 2024. URL https://sites.chemengr.ucsb.edu/~jbraw/mpc/MPC-book-2nd-edition-5th-printing.pdf. Second edition, fifth printing, September 2024. First printing of the second edition: October 2017.

David Silver and Richard S. Sutton. Welcome to the era of experience, 2025. Preprint chapter in *Designing an Intelligence*, MIT Press. Public preprint.

Jiawei Wang, Yushen Zuo, Ke Rui, Yichun Feng, and Minglei Li. Decision-metric alignment in latent world models: Diagnostics and action-conditioned objectives for MPC planning. *arXiv preprint arXiv:2608.18746*, 2026. URL https://arxiv.org/abs/2608.18746v1. Version 1, 19 August 2026.

Gaoyue Zhou, Hengkai Pan, Yann LeCun, and Lerrel Pinto. DINO-WM: World models on pre-trained visual features enable zero-shot planning. *arXiv preprint arXiv:2411.04983*, 2024. URL https://arxiv.org/abs/2411.04983v2. First posted 7 November 2024; version 2, 1 February 2025.