FEASIBLE ACTIONS, MISLEADING ROLLOUTS: DIAGNOSING LATENT WORLD-MODEL PLANNING

Anonymous authors
Paper under double-blind review

ABSTRACT

Prediction error, goal-cost disagreement, and actuator imprecision require different evidence in latent world-model planning. We study frozen planners by executing fixed action banks and comparing predicted cost, actual-outcome cost, and physical error. On ten V-JEPA2-AC/RoboCasa sources and ten LeWM/PushT tasks, predicted selection incurs mean range-normalized actual-cost regrets of 19.2% and 6.5%, respectively, while outperforming within-bank uniform selection. Seven LeWM banks select the actual-cost minimum; the distinct banks do not support a model ranking. We then test intermediate known-state feedback in DINO-WM. After a common actuator finish, feedback reduces composite object error relative to native planning, endpoint-only correction, and first-patch consistency control by 0.132, 0.090, and 0.098 on 32 generated tasks. Coarse-success non-inferiority remains unestablished. A finite-history analysis characterizes the learned state-error paths omitted from a shared comparison bound, without guaranteeing smaller realized errors. Separately, a 16-task LeWM terminal-constraint test does not confirm an object benefit. Together, the results show why candidate-level cost regret, physical outcomes, and actuator precision must be evaluated separately.

1 INTRODUCTION

World-model planning uses predictions to compare actions. A useful model therefore needs more than low average prediction error: it must preserve distinctions that matter to the selected action and to the task's physical outcome. Latent planners make this search practical, from PlaNet to DINO-WM, LeWorldModel, and V-JEPA2 (Hafner et al., 2019; Zhou et al., 2025; Maes et al., 2026; Assran et al., 2025). We examine frozen planners whose action sequences are ranked using distances between predicted and goal representations. Prediction is one component of a system containing a goal cost, an optimizer, and an execution rule.

A failed execution does not identify which component failed. An action can be predicted inaccurately, or its accurately represented outcome can receive a cost that disagrees with the physical objective. Conversely, a controller can achieve a more precise actuator endpoint without improving the object outcome. These alternatives require different evidence. Comparing predicted and executed costs measures prediction-induced selection loss on the same candidate support under the same cost. Comparing actual-outcome costs and physical errors tests agreement between the deployed objective and the specified physical metric. A controlled execution study then tests whether an intervention improves that metric.

Decision-focused learning evaluates predictions through downstream decisions (Mandi et al., 2022); uncertainty-aware control accounts for model error (Chua et al., 2018). Robot-Aware Control combines robot-specific dynamics with learned world prediction (Hu et al., 2022). Recent world-model studies examine proprioception, action consistency, physical supervision, and trajectory objectives (Terver et al., 2026; Seo et al., 2026; Yan et al., 2026; Huang et al., 2026). We combine these ingredients in a controlled audit of frozen planners: executed outcomes anchor the cost comparison, while endpoint, first-patch consistency, and common-finish controls distinguish consequences of supplying known self-state. We ask: **which action comparisons fail, and what object benefit remains when final actuator precision is controlled?**

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.

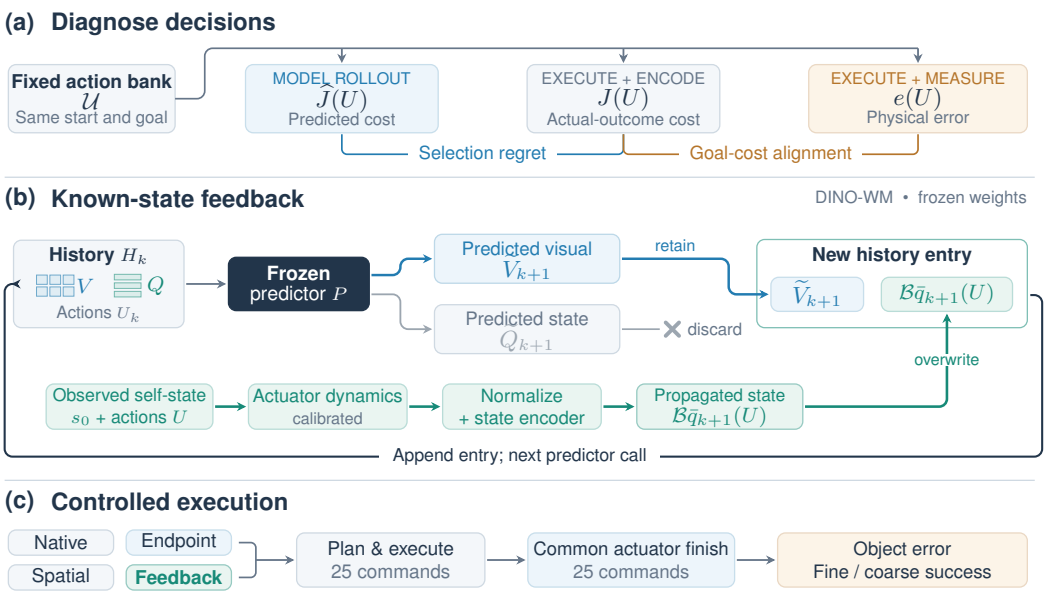


Figure 1: Diagnosis, known-state feedback, and controlled execution. (a) The same action bank is evaluated by predicted cost, actual-outcome cost, and physical error. (b) DINO-WM retains the predicted visual channels (blue), discards the predicted state channels (gray), and supplies a propagated state encoding (green). The new entry enters the retained history for the next predictor call; no future observed image is supplied. Arrows show computation, not measured causal effects. (c) All four DINO-WM arms receive the same actuator-finishing rule, then object error and fine/coarse success are evaluated. LeWM tests a terminal constraint, not this overwrite.

Two empirical findings organize the study. First, fixed-bank audits distinguish useful prediction-based selection from actual-cost regret and physical selection gaps; these quantities lead to different conclusions in the tested settings. Second, a four-arm common-finish study establishes a cohort-specific object benefit from supplying intermediate self-state to DINO-WM beyond endpoint-only correction and the tested first-patch consistency control. The primary object metric excludes actuator error; the benefit remains after a common actuator finish. Coarse-success non-inferiority is not established.

A finite-history analysis characterizes the computational effect of state replacement under explicit shared bounds. It does not predict the sign of realized performance changes. The separate LeWM terminal-constraint study tests the broader self-state-correction strategy, not the same overwrite operator. The diagnostic is applied in multiple settings, whereas the strongest controlled intervention evidence remains specific to DINO-WM.

2 Decision-Level Diagnosis

Fix the initial state, goal, execution horizon H, encoder, and native cost. For a legal action sequence $U = (u_0, \ldots, u_{H-1})$, let $y_H(U)$ and $\widehat{y}_H(U)$ denote actual and predicted terminal features:

$$J(U) = c(y_H(U), y_g), \qquad \widehat{J}(U) = c(\widehat{y}_H(U), y_g).$$

Physical error $e(U)$ is evaluated separately from simulator state. Actual features include proprioception when required by the interface. If a physically better candidate has higher actual cost, exact prediction preserves that disagreement. If predicted cost changes a preference present in actual cost, the discrepancy belongs to prediction under this fixed objective.

Common-bank selection. For each source s, fix a finite bank $\mathcal{U}_s$ before inspecting candidate outcomes and execute all candidates. With exact ties resolved by first stored index, define

$$U_p = \arg\min_{U \in \mathcal{U}_s} \widehat{J}(U), \quad U_a = \arg\min_{U \in \mathcal{U}_s} J(U), \quad R_{J,s} = J(U_p) - J(U_a) \geq 0. \tag{1}$$

Let $\Delta J_s = \max_{U \in \mathcal{U}_s} J(U) - \min_{U \in \mathcal{U}_s} J(U)$. We report source-wise range-normalized regret and the exact within-bank uniform-selection expectation:

$$r_s = \frac{J(U_p) - J(U_a)}{\Delta J_s}, \qquad r_{\mathrm{unif},s} = \frac{|\mathcal{U}_s|^{-1} \sum_{U \in \mathcal{U}_s} J(U) - J(U_a)}{\Delta J_s}. \tag{2}$$

Zero range maps to zero; positive ranges are not clipped. Raw-scale losses and near ties are reported in the supplement. Nonnegativity is definitional. Bank means describe the executed candidates; source-level intervals concern variation across sampled evaluation units. These quantities measure decision usefulness within the bank, not regret to an action-space optimum.

For $U_e = \arg\min_{U \in \mathcal{U}_s} e(U)$,

$$\underbrace{e(U_p) - e(U_e)}_{\text{total physical gap}} = \underbrace{e(U_p) - e(U_a)}_{\text{signed prediction difference}} + \underbrace{e(U_a) - e(U_e)}_{\text{actual-cost selection gap}}. \tag{3}$$

The signed term compares predicted-cost and actual-cost selection in physical units and can be negative. The last term measures the gap from actual-cost selection to the bank's physical minimum. These finite-set comparisons are not causal shares of a failure.

What the comparison identifies. The audit identifies differences between decision rules on the executed support, not the best action outside it. Pairwise reversals need not change the selected minimum. We therefore report selected indices, actual-cost regret, and physical outcomes, with pairwise counts as secondary diagnostics.

3 Known-State Feedback and Error Propagation

3.1 A controlled intervention in the rollout history

DINO-WM predicts visual patch channels V and repeated proprioceptive channels Q. For a fixed sequence U, let $\bar{q}_k(U)$ encode actuator state propagated from the observed initial state, and let $\mathcal{B}$ broadcast that encoding across patches. At prediction boundary k, the frozen predictor uses its retained history H_k and candidate command chunk U_k: $(\widetilde{V}_{k+1}, \widetilde{Q}_{k+1}) = P(H_k, U_k)$. Feedback then sets

$$(V_{k+1}, Q_{k+1}) \leftarrow (\widetilde{V}_{k+1}, \mathcal{B}\bar{q}_{k+1}(U)). \tag{4}$$

The current visual output is unchanged; the replacement enters subsequent calls through the retained history. Supplied state is exogenous to the learned recurrence but depends on U. Weights, state encoder, action interface, and search budget remain frozen.

The intervention requires a substate that can be propagated from available initial information and candidate actions without predicting the unobserved object state. In DINO-WM, calibrated actuator dynamics propagate pusher position and velocity at five-command boundaries. The checkpoint's normalization and state encoder map them into the newly appended history entry. Planning never substitutes an executed future image. The terminal cost retains the native visual/state dimensions and weights.

For a fixed action sequence, Endpoint, Spatial, and Feedback share the same propagated terminal-state term. Writing $\widehat{J}^m = \widehat{J}^m_v + \widehat{J}_q$ gives

$$\widehat{J}^{\mathrm{Feedback}}(U) - \widehat{J}^{\mathrm{Endpoint}}(U) = \widehat{J}^{\mathrm{Feedback}}_v(U) - \widehat{J}^{\mathrm{Endpoint}}_v(U). \tag{5}$$

Table 1: DINO-WM rollout controls. For a fixed sequence, the three corrections share a propagated terminal state cost. Every arm in the common-finish experiment receives the same finishing rule. Endpoint changes terminal scoring; Spatial broadcasts the first predicted patch state.

Planner	Intermediate proprioceptive input	Terminal proprioceptive cost
Native	Original predicted patch channels	Original predicted state
Endpoint	Original predicted patch channels	Propagated state
Spatial	Predicted first-patch state broadcast	Propagated state
Feedback	Propagated state broadcast	Propagated state

Their cost differences on that sequence are entirely in the predicted visual term. This equality does not equate costs of different CEM-selected sequences. Spatial broadcasts the first predicted patch state, not a patch mean.

Feedback versus Endpoint tests changing intermediate predictor inputs beyond terminal substitution. Feedback versus Spatial tests supplying the propagated value beyond enforcing this specific patch agreement. Reoptimized plans can differ across arms. Their endpoint comparison evaluates the complete planning intervention; the post hoc shared-candidate audit measures a scoring consequence. Neither identifies mediation by contact-prediction accuracy.

3.2 Finite-history error propagation

Fix U and K prediction boundaries (five boundaries for 25 commands here). Stack visual feature-error norms into $\mathbf{e}_V \in \mathbb{R}_+^K$ and state-channel error norms into $\mathbf{e}_Q$; these differ from physical e_{obj}. Let $\boldsymbol{\eta}_V, \boldsymbol{\eta}_Q$ be predictor residual norms on actual histories. With the same exactly observed initial history, assume common nonnegative, strictly lower triangular matrices A, B, C, D bound dependence on retained past errors:

$$\mathbf{e}_{V,\mathrm{free}} \le \boldsymbol{\eta}_V + A\mathbf{e}_{V,\mathrm{free}} + B\mathbf{e}_{Q,\mathrm{free}}, \qquad \mathbf{e}_{Q,\mathrm{free}} \le \boldsymbol{\eta}_Q + C\mathbf{e}_{V,\mathrm{free}} + D\mathbf{e}_{Q,\mathrm{free}}. \tag{6}$$

The constants cover actual, free, and feedback histories on a shared domain. Entries vanish for unretained positions; no one-step Markov approximation is imposed. Define supplied-state error $\xi_j = \|\mathcal{B}\bar{q}_j(U) - Q_j^*\|$, stacked into $\boldsymbol{\xi}$.

Proposition 1 (Finite-history comparison under state replacement)**.** *Under these assumptions, let $M = A + B(I - D)^{-1}C$. Componentwise,*

$$\mathbf{e}_{V,\mathrm{free}} \le (I - M)^{-1}\big[\boldsymbol{\eta}_V + B(I - D)^{-1}\boldsymbol{\eta}_Q\big], \tag{7}$$

$$\mathbf{e}_{V,\mathrm{Feedback}} \le (I - A)^{-1}(\boldsymbol{\eta}_V + B\boldsymbol{\xi}). \tag{8}$$

If $\boldsymbol{\xi} = 0$, the right-hand side of Equation 8 is no greater than the right-hand side of Equation 7.

The bound omits recursively generated state residuals and the visual-to-state-to-visual comparison term. Supplied-state error and visual recurrence remain. Endpoint changes only terminal scoring; Feedback changes recursive history. Appendix A proves the result using finite nonnegative inverses, without contraction. The coefficients are not estimated, and calibrated dynamics do not establish $\boldsymbol{\xi} = 0$.

Shared constants must cover every compared history. Visual recurrence and supplied-state error can still dominate; realized error and physical performance are not ordered.

Error cancellation can reverse the benefit. Let true visual and state values be zero, and let a predictor output $\widehat{v}_{k+1} = \widehat{q}_k + 1$, $\widehat{q}_{k+1} = -1$. From exact initial state, free rollout has visual error zero at step two, whereas exact state replacement produces error one. Replacement removes a cancellation between errors (Appendix A).

3.3 From differential cost error to action selection

Define $\epsilon(U) = \widehat{J}(U) - J(U)$ on a fixed finite bank $\mathcal{U}$.

Lemma 2 (Finite-bank selection bound). *For $\widehat{U} \in \mathcal{U}$ satisfying $\widehat{J}(\widehat{U}) \leq \min_{U\in\mathcal{U}} \widehat{J}(U) + \eta_{\mathrm{opt}}$, with $\eta_{\mathrm{opt}} \geq 0$,*

$$J(\widehat{U}) - \min_{U\in\mathcal{U}} J(U) \leq \eta_{\mathrm{opt}} + \sup_{U\in\mathcal{U}} \epsilon(U) - \inf_{U\in\mathcal{U}} \epsilon(U). \tag{9}$$

A common additive cost error cancels; selection depends on differences in cost error across candidates. The proof and squared-distance expansion appear in Appendix A. Exact bank selection has $\eta_{\mathrm{opt}} = 0$; a final CEM mean has no such guarantee. This independent decision bound complements Proposition 1: rollout error matters for selection through the resulting candidate costs. We do not derive a uniform cost-error bound from the history comparison.

4 Evaluation Protocol

Systems and banks. We use frozen DINO-WM and LeWM on PushT, and V-JEPA2-AC in a RoboCasa reaching diagnostic using robosuite (Nasiriany et al., 2024; Zhu et al., 2025). Each bank contains zero action and 15 sampled sequences. RoboCasa uses its frozen candidate generator and executes three commands over 15 physical steps. LeWM samples its existing initial standard-normal proposal distribution, applies its scaler and legal clipping, and executes 25 commands. Neither bank is optimized by CEM or uniformly sampled from the action space. Uniform selection is a within-bank reference. Different goal constructions, horizons, and costs preclude a model ranking.

RoboCasa uses the retrained no-proprioception configuration from the JEPA-WM design study (Terver et al., 2026), with its frozen action-conditioned predictor and the original V-JEPA2 encoder (Appendix F.3). All RoboCasa results refer to this checkpoint and interface.

Functional controls. DINO-WM uses 32 generated tasks, two optimizer seeds, and the four arms in Table 1. LeWM uses the predetermined first 16 tasks, two seeds, and Native versus its existing terminal-state constraint. Each arm executes a 25-command CEM plan (300 samples, 30 elites, 30 updates) and a 25-command analytic finish. LeWM corrects terminal scoring; it does not replicate DINO-WM's overwrite operator. Both use frozen RGB-calibrated actuator gains, $k_p = 102.1442, k_d = 20.4151$ (Appendix A.7); the current/goal self-state and action conventions are specified in Table 9.

All arms use the same observation interface and schedule. Each finish is computed from that arm's own observed command-25 state toward the shared goal state, solving the actuator endpoint problem. Finishing actions and object contacts can differ. The comparison tests object outcomes of these two-stage controllers after final actuator precision is constrained.

Tasks and endpoints. PushT uses the public simulator associated with Diffusion Policy (Chi et al., 2024). The generator samples object center and orientation, places the pusher behind it, and constructs a reachable goal with 20 fixed-magnitude pushes followed by five zero commands. All 32 sampled tasks require object motion and initially fail both endpoint criteria; no source is replaced or filtered using model outcomes. A task is the complete initial-state–goal pair, not an optimizer run. This controlled construction differs from the standard silhouette-coverage benchmark.

For pusher position p, object position o, and angle θ, define

$$e_{\mathrm{obj}} = \max\left\{ \frac{\|o - o_g\|_2}{20\,\mathrm{px}}, \frac{|\,\mathrm{wrap}(\theta - \theta_g)|}{20^\circ} \right\}, \tag{10}$$

$$e_{\mathrm{joint}} = \max\left\{ \frac{\sqrt{\|p - p_g\|_2^2 + \|o - o_g\|_2^2}}{20\,\mathrm{px}}, \frac{|\,\mathrm{wrap}(\theta - \theta_g)|}{20^\circ} \right\}. \tag{11}$$

Fine/coarse success require $e_{\mathrm{joint}} < 0.25/1$. Velocity is a separate tolerance check. Every execution in the main common-finish comparisons ends within 1 pixel and 1 pixel/s of the goal actuator state, without exclusions. RoboCasa uses hand–object distance and a 0.2 m terminal criterion; the separate goal-semantic audit uses EEF-goal position. Neither denotes completion of a kitchen manipulation task.

Inference and analysis status. The primary post-finish metric is command-50 object error. Two seeds are averaged within task before 10,000 task-bootstrap draws; intervals are pointwise 95%. Original Feedback-minus-Endpoint and added Feedback-minus-Spatial comparisons have separately fixed primary roles; Feedback-minus-Native is secondary. LeWM's primary is Constraint minus Native. LeWM bank tasks are a subset of its functional tasks, themselves a subset of the evaluated DINO-WM cohort. RoboCasa bank construction preceded outcome inspection; subsequent regret and physical-decomposition analyses are post hoc. Paired sign flips require a sign-exchangeable null: we enumerate signs for ten/16 units and use 50,000 Monte Carlo flips for 32. For definitionally nonnegative gaps, we retain descriptive means, bootstrap summaries, and original sign-flip values without treating the latter as assumption-free tests of a positive population mean (Appendix E.2).

5 Results

5.1 Fixed-candidate decisions and physical consequences

Useful selection with remaining cost regret. On ten RoboCasa banks, normalized regret is 19.24% [6.18,34.25], versus 49.86% [42.38,56.47] for uniform selection; the paired difference is -30.62 percentage points [-46.05,-14.02]. On ten LeWM banks, regret is 6.53% [0,15.58], versus 61.46% [55.09,66.64]; the paired difference is -54.93 points [-64.37,-44.23], sign-flip $p = .00195$. Prediction is useful in both banks. In LeWM, nonzero regret occurs in three of ten banks (Figure 2). Its smaller observed mean is not a cross-model performance comparison.

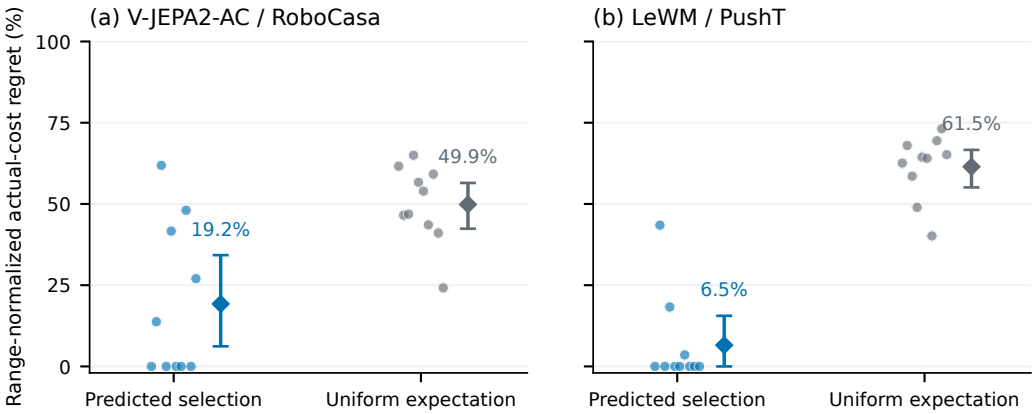


Figure 2: Prediction-induced actual-cost regret on separately evaluated banks. Points show all ten source values and each bank's exact uniform-selection expectation; LeWM has 7/10 exact zero regrets. Zero values are displaced horizontally only. Diamonds and bars show means and pointwise source-bootstrap 95% intervals. Both panels use the same scale; different systems, goals, and horizons prevent a cross-model ranking.

Actual-cost selection and physical outcomes. In RoboCasa, selecting by actual rather than predicted cost does not establish a physical improvement: the signed prediction difference is +0.01950 m [-0.01690,+0.05833]. Actual-cost selection still has a 0.07621 m [0.04914,0.10396] gap to the bank's physical minimum. Predicted selection improves physical distance over uniform selection by 0.05135 m [0.02246,0.07857].

In LeWM, the finite-bank physical gap is not accompanied by evidence of harm from predicted-cost selection relative to actual-cost selection. The signed difference is -0.00308 [-0.08259,+0.06661] object-error units; the actual-cost selection gap is 0.07744 [0.01057,0.15122]. These sum to the observed total gap of 0.07436. Actual-cost and physical minima coincide on six tasks. With only four nonzero nonnegative gaps, the original two-sided sign-flip value is .125; discreteness and its null assumptions differ from the bootstrap summary (Appendix E.2).

Local reversals are a different endpoint. Among 160 eligible matched pairs from ten RoboCasa sources, ten actual-cost/physical reversals occur across five sources. Yet minima agree in all ten

unfiltered seven-candidate banks, always at the largest tested translation scale, 1.5. This bank and EEF-goal position metric differ from the bank above; their agreement is limited to the tested support. The fingerpad measurement was revised after acquisition, with the failed original analysis retained (Appendix F.31.6).

Earlier outcome-conditioned witness comparisons show 44/46 LeWM and 37/38 DINO-WM reversals. Actual intermediate inputs change these rankings but require future observations. In RoboCasa's fixed bank, actual-prefix substitution does not establish reduced selection regret. The fixed banks quantify selection loss without outcome-based pair filtering; the two designs do not estimate the same error rate.

5.2 Object benefit after common actuator finishing

Table 2: Common-finish outcomes. Object error is the dimensionless composite e_{obj} at command 50. Counts are executions; paired inference averages seeds within task. Different cohorts and correction interfaces preclude a model ranking.

System (tasks × seeds)	Planner	$e_{\mathrm{obj}}(50) \downarrow$	Fine	Coarse
DINO-WM (32×2)	Native	.47643	21/64	59/64
	Endpoint	.43452	22/64	61/64
	Spatial	.44279	22/64	62/64
	Feedback	.34439	37/64	59/64
LeWM (16×2)	Native	.72304	11/32	26/32
	Terminal constraint	.70058	13/32	26/32

Feedback improves the object endpoint beyond the tested controls. With $\Delta = \mathrm{Feedback} - \mathrm{Control}$, command-50 object differences are -0.13204 [-0.20885,-0.06477] versus Native, -0.09012 [-0.15421,-0.01902] versus Endpoint, and -0.09840 [-0.17263,-0.01748] versus Spatial (Table 2; Figure 4). The primary metric contains no pusher-position or velocity term, and the benefit remains after all arms satisfy the same final actuator tolerances. The intervals support the direction of improvement but do not establish a benefit of at least the original 0.1-unit meaningful-effect reference. Spatial is an added matched control on the observed cohort, not an independent replication.

Benefit precedes finishing. Feedback minus Spatial is -0.16764 [-0.24835,-0.08139] at command 25 and -0.09840 at command 50. The common finish attenuates this advantage rather than creating it. Figure 3 connects the intervention, a recorded-state example, and these cohort effects. This stage decomposition is descriptive because trajectories and contacts differ (Appendix F.31.5).

Shared-candidate scoring has a selection-source limitation. Post hoc cross-scoring compares overlapping banks of three optimized proposals and four after adding Spatial. Feedback improves actual-cost regret against Spatial in both. The selected command-50 physical contrast crosses zero in the three-candidate bank (-0.04290 [-0.09953,+0.02129]) but excludes zero in the four-candidate bank (-0.10174 [-0.17473,-0.02215]). Feedback selects its own optimized proposal in 64/64 and 61/64 banks, respectively. These scoring results largely reuse the original method's physical outcomes; they are not a second independent functional confirmation or evidence of a unique contact-prediction mediator.

Fine success reaches 37/64 versus 22/64 for Spatial: 17 paired gains and two losses. Coarse success is 59/64 versus 62/64: no gains and three losses. Coarse-success non-inferiority is not established. Position and angle component intervals each cross zero; the improvement belongs to the specified maximum-normalized composite, not to separately confirmed gains in both components. The supplement retains threshold curves, all task means, and the earlier failed overall criteria.

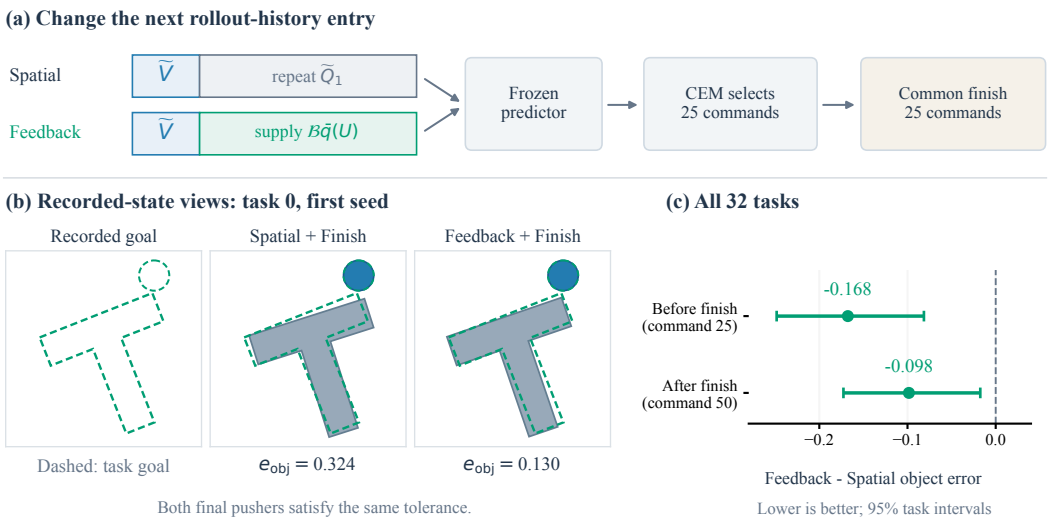


Figure 3: Feedback mechanism and object benefit. (a) The state part of the new rollout-history entry changes; visual channels remain predicted. Both controls share terminal-state scoring and the finishing rule. (b) Saved-state renderings of the first stored task/seed pair, without effect-based selection. Dashed outlines show the recorded task goal; circles are the pusher. No new simulation is performed. (c) Feedback-minus-Spatial object error across all 32 tasks, averaging two seeds before task-bootstrap 95% intervals. Benefit precedes and remains after finishing.

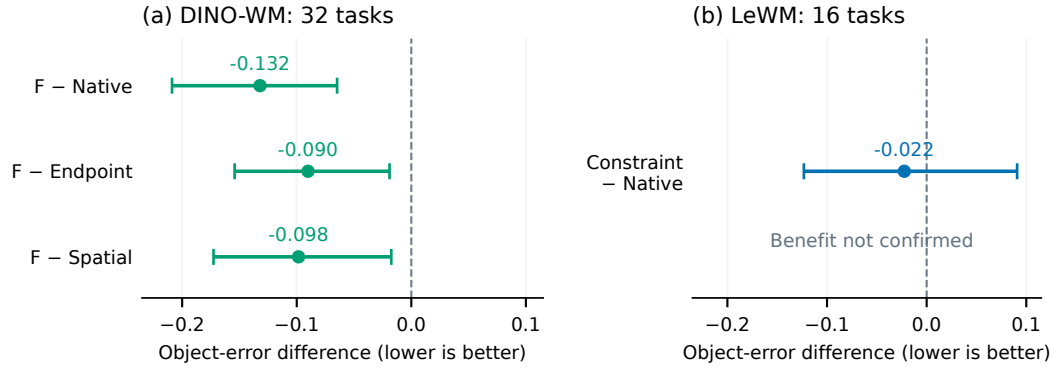


Figure 4: Post-finish object effects across correction interfaces. Left: Feedback minus each DINO-WM control on 32 tasks; F–E is the original primary comparison, F–S the added fixed comparison, and F–N secondary. Right: LeWM Terminal constraint minus Native on 16 predetermined tasks. Lower is better; bars are pointwise task-bootstrap 95% intervals after averaging seeds. Interventions and cohorts differ; the LeWM interval establishes neither benefit nor equivalence.

5.3 Scope across cohorts and correction interfaces

LeWM does not confirm a post-finish object benefit. Terminal constraint minus Native is -0.02246 [-0.12326,+0.09080], sign-flip $p = .704$, on 16 tasks. Fine success is 13/32 versus 11/32; coarse success is 26/32 in both arms. At command 25, joint error improves (-0.30079 [-0.46409,-0.12175]), but object error does not show a confirmed improvement. After finishing, every actuator meets the tolerance, while the object interval includes both lower and higher mean error. This is an unconfirmed benefit, not equivalence or an experiment on the overwrite operator.

Earlier ten reserved temporal windows gave a positive LeWM result (-0.24158 [-0.51402,-0.02786]) and an unconfirmed DINO-WM result (+0.05825 [-0.01104,+0.15513]). Those windows share source episodes and differ from the generated tasks. Significance in one cohort and nonsignifi-

cance in another does not prove a cohort interaction. The interface and intervention differences also prevent an architectural explanation. Additional RoboCasa scoring-time controls do not establish improved terminal reaching (Appendix F.31.7).

5.4 Sensitivity to imperfect supplied states

The supplied-state term in Equation 8 motivates testing approximate state providers. We rescore the existing DINO-WM four-candidate banks from 32 tasks and two seeds under 87 fixed conditions, perturbing future supplied position, velocity, or calibrated dynamics. Feedback, Endpoint, and Spatial share the perturbed terminal-state cost; initial observations and goals remain unchanged. These are 22,272 candidate-condition scores, with no new CEM search or simulation (Appendix G).

At 5 simulation-coordinate pixels of persistent position bias or boundary-jitter RMS, Feedback-minus-Spatial actual-cost regret is -0.0617/-0.0601. The selected candidates' command-50 object differences are -0.0791 [-0.1460,-0.0145] and -0.0863 [-0.1594,-0.0095], respectively. Approximate bootstrap bands across tested strengths within each error family remain negative for both endpoints. At 10 pixels, both object intervals cross zero. The 5-pixel jitter comparison against Endpoint also crosses zero; benefit is not established against every control. Only 60.2%/77.3% of nominal Feedback selections are retained at 5 pixels, so retained average benefit does not mean unchanged decisions.

Velocity perturbations retain a negative object contrast against Spatial through the tested 80 pixels/s. Joint gain changes of up to $\pm 20\%$ retain 98.4–100% of selections but induce less than 0.7 pixels of mean additional position RMS. These results motivate approximate, action-conditioned self-state providers with component-specific accuracy checks. The practical implementation and evaluation of such providers remain future work.

6 Related Work and Limitations

Decision-focused learning trains predictions for downstream decisions (Mandi et al., 2022); we audit frozen checkpoints using executed banks. Robot-Aware Control separates robot-specific and learned world dynamics (Hu et al., 2022); we instead write to an existing frozen state interface and test endpoint and first-patch controls. Analytic constraints are established tools (Beucler et al., 2021); our analysis describes their finite-history consequences. DINO-WM uses DINOv2 features (Oquab et al., 2024); adjacent work changes representations, supervision, trajectory costs, or scoring time (Fu et al., 2026; Yan et al., 2026; Huang et al., 2026; Vesessook & Yang, 2026). We do not infer superiority over unevaluated methods.

Three conditions limit scope. The overwrite requires a writable state channel and a substate propagatable without unknown object dynamics; LeWM uses a different interface. Evidence concerns specific checkpoints, simulators, banks, and generated tasks, which are not certified pretraining OOD. Finally, neither the physical comparisons nor the conditional theory identifies contact-prediction mediation, closed-loop stability, or uniform reliability. Common finishing constrains final actuator error while allowing different contact histories.

7 Conclusion

Executed candidate banks reveal whether prediction changes the deployed objective's decisions and whether those decisions align with the chosen physical metric. In DINO-WM, intermediate known-state feedback improves object outcomes beyond endpoint and first-patch controls under common finishing. LeWM and coarse-success results delimit this functional claim. Finite-history analysis characterizes altered error paths but does not replace empirical evaluation. World-model interventions should therefore be assessed at the levels of candidate selection, actual-outcome cost, and controlled physical execution.

REPRODUCIBILITY STATEMENT

The anonymous package includes configurations, source identities, compact candidate-cost and endpoint arrays, CPU analyses, figures, and manuscript sources. These reconstruct the reported selections, statistical summaries, and plots. New LeWM data include full saved feature and trajectory records; several earlier analyses instead include compact arrays and original raw-feature/trajectory audit records, not all underlying tensors. Full model or simulator reruns require external checkpoints, assets, and the documented environments. Table 22 maps results to entries and specifies these boundaries. Earlier failures and measurement revisions remain available. Numerical audits of saved data are distinct from independent experimental replication.

AI USE STATEMENT

In this work, we used generative AI tools for generating synthetic data sets, helping refine theoretical models, formulating mathematical claims, providing critical ingredients for proving mathematical claims, assisting in the writing of proofs, implementing methods, drafting parts of the manuscript, and interpreting some of the results. AI tools also assisted with proposing and refining hypotheses and designing or providing feedback on research methodology and experiments, including the supplied-state sensitivity study. Additionally, we used generative AI tools for creating or modifying figures and tables, suggesting experimental parameters, summarizing existing literature, and polishing the manuscript. Synthetic evaluation tasks and experimental measurements were produced by program execution and saved outputs, not by fabricated observations. We take responsibility for the final content of this work, including text, claims or artifacts produced with the aid of generative AI.

REFERENCES

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

Tom Beucler, Michael Pritchard, Stephan Rasp, Jordan Ott, Pierre Baldi, and Pierre Gentine. Enforcing analytic constraints in neural networks emulating physical systems. *Physical Review Letters*, 126(9):098302, March 2021. ISSN 1079-7114. doi: 10.1103/physrevlett.126.098302. URL http://dx.doi.org/10.1103/PhysRevLett.126.098302.

Stephen Boyd and Lieven Vandenberghe. *Introduction to Applied Linear Algebra: Vectors, Matrices, and Least Squares*. Cambridge University Press & Assessment, 2018. ISBN 9781316518960. doi: 10.1017/9781108583664. URL http://dx.doi.org/10.1017/9781108583664.

Cheng Chi, Zhenjia Xu, Siyuan Feng, Eric Cousineau, Yilun Du, Benjamin Burchfiel, Russ Tedrake, and Shuran Song. Diffusion policy: Visuomotor policy learning via action diffusion, 2024. URL https://arxiv.org/abs/2303.04137.

Kurtland Chua, Roberto Calandra, Rowan McAllister, and Sergey Levine. Deep reinforcement learning in a handful of trials using probabilistic dynamics models, 2018. URL https://arxiv.org/abs/1805.12114.

Pieter-Tjerk de Boer, Dirk P. Kroese, Shie Mannor, and Reuven Y. Rubinstein. A tutorial on the cross-entropy method. *Annals of Operations Research*, 134(1):19–67, February 2005. ISSN 1572-9338. doi: 10.1007/s10479-005-5724-z. URL http://dx.doi.org/10.1007/s10479-005-5724-z.

B. Efron. Bootstrap methods: Another look at the jackknife. *The Annals of Statistics*, 7(1):1–26, January 1979. ISSN 0090-5364. doi: 10.1214/aos/1176344552. URL http://dx.doi.org/10.1214/aos/1176344552.

Minghao Fu, Fan Feng, Nicklas Hansen, and Biwei Huang. Back to parsimonious latents: Learning task-centric world models from visual foundations, 2026. URL https://arxiv.org/abs/2605.25620.

Danijar Hafner, Timothy Lillicrap, Ian Fischer, Ruben Villegas, David Ha, Honglak Lee, and James Davidson. Learning latent dynamics for planning from pixels. In Kamalika Chaudhuri and Ruslan Salakhutdinov (eds.), *Proceedings of the 36th International Conference on Machine Learning*, volume 97 of *Proceedings of Machine Learning Research*, pp. 2555–2565. PMLR, 09–15 Jun 2019. URL https://proceedings.mlr.press/v97/hafner19a.html.

Jesse Hemerik and Jelle Goeman. Exact testing with random permutations. *TEST*, 27(4):811–825, 2018. ISSN 1863-8260. doi: 10.1007/s11749-017-0571-1. URL http://dx.doi.org/10.1007/s11749-017-0571-1.

Edward S. Hu, Kun Huang, Oleh Rybkin, and Dinesh Jayaraman. Know thyself: Transferable visual control policies through robot-awareness, 2022. URL https://arxiv.org/abs/2107.09047.

Xiaodi Huang, Ziyi Ding, Jingtian Wan, Yuchen Liu, Yuan Zhang, Xiao-Ping Zhang, Jiayu Chen, Zhang Zhang, and Tao Huang. Traj-LeWM: Path-aware world-model planning via latent trajectory cost, 2026. URL https://arxiv.org/abs/2608.14125.

Alexander Khazatsky, Karl Pertsch, Suraj Nair, Ashwin Balakrishna, Sudeep Dasari, Siddharth Karamcheti, Soroush Nasiriany, Mohan Kumar Srirama, Lawrence Yunliang Chen, Kirsty Ellis, Peter David Fagan, Joey Hejna, Masha Itkina, Marion Lepert, Yecheng Jason Ma, Patrick Tree Miller, Jimmy Wu, Suneel Belkhale, Shivin Dass, Huy Ha, Arhan Jain, Abraham Lee, Youngwoon Lee, Marius Memmel, Sungjae Park, Ilija Radosavovic, Kaiyuan Wang, Albert Zhan, Kevin

Black, Cheng Chi, Kyle Beltran Hatch, Shan Lin, Jingpei Lu, Jean Mercat, Abdul Rehman, Pannag R Sanketi, Archit Sharma, Cody Simpson, Quan Vuong, Homer Rich Walke, Blake Wulfe, Ted Xiao, Jonathan Heewon Yang, Arefeh Yavary, Tony Z. Zhao, Christopher Agia, Rohan Baijal, Mateo Guaman Castro, Daphne Chen, Qiuyu Chen, Trinity Chung, Jaimyn Drake, Ethan Paul Foster, Jensen Gao, David Antonio Herrera, Minho Heo, Kyle Hsu, Jiaheng Hu, Donovon Jackson, Charlotte Le, Yunshuang Li, Roy Lin, Zehan Ma, Abhiram Maddukuri, Suvir Mirchandani, Daniel Morton, Tony Nguyen, Abigail O'Neill, Rosario Scalise, Derick Seale, Victor Son, Stephen Tian, Emi Tran, Andrew E. Wang, Yilin Wu, Annie Xie, Jingyun Yang, Patrick Yin, Yunchu Zhang, Osbert Bastani, Glen Berseth, Jeannette Bohg, Ken Goldberg, Abhinav Gupta, Abhishek Gupta, Dinesh Jayaraman, Joseph J Lim, Jitendra Malik, Roberto Martín-Martín, Subramanian Ramamoorthy, Dorsa Sadigh, Shuran Song, Jiajun Wu, Michael C. Yip, Yuke Zhu, Thomas Kollar, Sergey Levine, and Chelsea Finn. DROID: A Large-Scale In-The-Wild Robot Manipulation Dataset. In *Proceedings of Robotics: Science and Systems*, Delft, Netherlands, July 2024. doi: 10.15607/RSS.2024.XX.120.

Lucas Maes, Quentin Le Lidec, Damien Scieur, Yann LeCun, and Randall Balestriero. LeWorldModel: Stable end-to-end joint-embedding predictive architecture from pixels. https://arxiv.org/abs/2603.19312, 2026. v3.

Jayanta Mandi, Víctor Bucarey, Maxime Mulamba Ke Tchomba, and Tias Guns. Decision-focused learning: Through the lens of learning to rank. In *Proceedings of the 39th International Conference on Machine Learning*, volume 162 of *Proceedings of Machine Learning Research*, pp. 14935–14947. PMLR, 2022. URL https://proceedings.mlr.press/v162/mandi22a.html.

Soroush Nasiriany, Abhiram Maddukuri, Lance Zhang, Adeet Parikh, Aaron Lo, Abhishek Joshi, Ajay Mandlekar, and Yuke Zhu. RoboCasa: Large-scale simulation of everyday tasks for generalist robots, 2024. URL https://arxiv.org/abs/2406.02523.

Maxime Oquab, Timothée Darcet, Théo Moutakanni, Huy 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, 2024. URL https://arxiv.org/abs/2304.07193.

Gawon Seo, Dongwon Kim, and Suha Kwak. ACID: Action consistency via inverse dynamics for planning with world models, 2026. URL https://arxiv.org/abs/2607.02403.

Basile Terver, Tsung-Yen Yang, Jean Ponce, Adrien Bardes, and Yann LeCun. What drives success in physical planning with joint-embedding predictive world models?, 2026. URL https://arxiv.org/abs/2512.24497.

Emanuel Todorov, Tom Erez, and Yuval Tassa. Mujoco: A physics engine for model-based control. In *2012 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pp. 5026–5033. IEEE, October 2012. doi: 10.1109/iros.2012.6386109. URL http://dx.doi.org/10.1109/IROS.2012.6386109.

Kanpat Vesessook and Kevin Yang. Hidden failure modes in latent world-model planning from offline data. In *ICML 2026 Workshop on Decision-Making from Offline Datasets to Online Adaptation*, 2026. URL https://github.com/24GUNV/LeWMRO.

Haodong Yan, Jiaguan Zhu, Mingyuan Jia, Ruiqing Yin, Junjie He, Zhide Zhong, Junfeng Li, Jinxuan Lu, Hengtao Li, Tianran Zhang, Jiayi Chen, Wenxuan Song, Wen Chen, Yuxiang Gao, and Haoang Li. Is forward prediction enough? physical state grounding for JEPA world models, 2026. URL https://arxiv.org/abs/2608.06799.

Gaoyue Zhou, Hengkai Pan, Yann LeCun, and Lerrel Pinto. DINO-WM: World models on pre-trained visual features enable zero-shot planning. https://arxiv.org/abs/2411.04983, 2025. v2.

Yuke Zhu, Josiah Wong, Ajay Mandlekar, Roberto Martín-Martín, Abhishek Joshi, Kevin Lin, Abhiram Maddukuri, Soroush Nasiriany, and Yifeng Zhu. robosuite: A modular simulation framework and benchmark for robot learning, 2025. URL https://arxiv.org/abs/2009.12293.

SUPPLEMENTARY NAVIGATION

The main-text results can be read directly through the following entries. Earlier cohorts and failed criteria retain their original scope and are not pooled with the generated tasks. The stand-alone supplementary PDF appends the shared reference list.

Evidence	Location
DINO-WM four-arm common finish	Appendices F.31.3–F.31.5
RoboCasa fixed-bank decisions	Appendices F.31.1–F.31.2
New LeWM bank and functional comparison	Appendix E
Supplied-state error sensitivity	Appendix G
Finite-history analysis and selection bound	Appendix A
Earlier cohorts and failed complete criteria	Appendix F.6 and subsequent cohort-specific sections
Current reproduction entries	Appendix F.32

A FINITE-HISTORY ANALYSIS AND IDENTIFICATION LIMITS

These results formalize an implemented state replacement and the finite-bank comparisons. The matrix identities, affine projection, and decision bound are elementary. They do not establish a new generalization theorem or identify the mechanism of an observed performance gain. All inequalities between vectors or matrices below are componentwise unless a norm is explicitly written.

A.1 ASSUMPTIONS AND FULL-HISTORY NOTATION

Fix a legal action sequence U and a horizon of K prediction boundaries. The actual features at boundary j are V_j^*, Q_j^*, with Q_j^* in the same patch-broadcast state interface used by the model. The analysis assumes the same exactly observed initial feature history. This is a reference condition, not a claim that RGB state estimation in the experiments is noise-free. Nonzero initial feature errors would add forcing terms to the comparison inequalities; the displayed bounds use the stated zero-initial-error case. Let $\mathcal{H}_j \subseteq \{1, \dots, j-1\}$ identify the predicted entries retained for call j; any initially observed entries have zero comparison error. The predictor can use the entire admitted history, its positional encoding, and candidate actions. We do not replace this computation by a first-order Markov model.

Write the frozen predictor's outputs as $P_{V,j}(H, U)$ and $P_{Q,j}(H, U)$. Define residuals on the actual history H_j^* by

$$\eta_{V,j} = \|P_{V,j}(H_j^*, U) - V_j^*\|, \qquad \eta_{Q,j} = \|P_{Q,j}(H_j^*, U) - Q_j^*\|.$$

Choose any fixed norm on each full visual and state block; the coefficients below are defined for those choices. The residuals above are norms, not signed vectors. Assume that common nonnegative coefficients $A_{ji}, B_{ji}, C_{ji}, D_{ji}$, zero for $i \notin \mathcal{H}_j$, satisfy

$$\|P_{V,j}(H, U) - P_{V,j}(H_j^*, U)\| \le \sum_{i<j} \big(A_{ji}\|V_i - V_i^*\| + B_{ji}\|Q_i - Q_i^*\|\big), \tag{12}$$

$$\|P_{Q,j}(H, U) - P_{Q,j}(H_j^*, U)\| \le \sum_{i<j} \big(C_{ji}\|V_i - V_i^*\| + D_{ji}\|Q_i - Q_i^*\|\big) \tag{13}$$

for every history being compared, including the free and feedback histories. Thus the domain and constants are shared across interventions. These inequalities may be obtained from suitable uniform Lipschitz bounds on a common region, but a Jacobian at only the clean history is not enough to establish them. No constants are estimated in our experiments.

Let $e_{V,j} = \|\widehat{V}_j - V_j^*\|$ and $e_{Q,j} = \|\widehat{Q}_j - Q_j^*\|$. The triangle inequality yields Equation 6. The matrices are strictly lower triangular because a prediction depends only on earlier history entries. Feedback stores $\mathcal{B}\bar{q}_j(U)$ instead of the predicted state output. Define

$$\xi_j = \|\mathcal{B}\bar{q}_j(U) - Q_j^*\|.$$

This compares propagated and actual state through the same fixed encoding and broadcast interface. An incorrect propagated physical state can yield an encoded error, whose size depends on that interface; identical physical states under the same deterministic encoding do not create an additional representation error. Physical position tolerance alone does not establish a numerical bound on $\boldsymbol{\xi}$ without a suitable encoder bound. Exact supplied state, $\boldsymbol{\xi} = 0$, is a theoretical case and is not assumed for the calibrated experiments. Supplying state does not replace the visual output of that same call.

A.2 Proof of Proposition 1

For a nonnegative strictly lower triangular $K \times K$ matrix L, $L^K = 0$. Consequently,

$$(I - L)^{-1} = I + L + \cdots + L^{K-1} \geq 0.$$

This identity holds even when an induced norm of L exceeds one. It is a finite-horizon fact, not a contraction or stability result. Applying it to the state inequality gives

$$\mathbf{e}_{Q,\text{free}} \leq (I - D)^{-1}(\boldsymbol{\eta}_Q + C\mathbf{e}_{V,\text{free}}).$$

Substituting into the visual inequality produces

$$\mathbf{e}_{V,\text{free}} \leq \boldsymbol{\eta}_V + B(I - D)^{-1}\boldsymbol{\eta}_Q + \underbrace{\left[A + B(I - D)^{-1}C\right]}_{M} \mathbf{e}_{V,\text{free}}.$$

The matrix M is also nonnegative and strictly lower triangular. Multiplication by its nonnegative finite inverse proves Equation 7. In the feedback rollout, the stored state error is exactly $\boldsymbol{\xi}$; the learned state output is discarded. Thus

$$\mathbf{e}_{V,\text{Feedback}} \leq \boldsymbol{\eta}_V + A\mathbf{e}_{V,\text{Feedback}} + B\boldsymbol{\xi},$$

which proves Equation 8. If $\boldsymbol{\xi} = 0$, then $M \geq A \geq 0$, implying $M^j \geq A^j$ by induction. Therefore

$$(I - M)^{-1} \geq (I - A)^{-1}, \qquad \boldsymbol{\eta}_V + B(I - D)^{-1}\boldsymbol{\eta}_Q \geq \boldsymbol{\eta}_V.$$

Multiplying nonnegative quantities proves the stated ordering of upper bounds. No ordering of realized errors follows from two different upper bounds. With nonzero $\boldsymbol{\xi}$, even this upper-bound ordering needs an additional condition, for example $B\boldsymbol{\xi} \leq B(I - D)^{-1}\boldsymbol{\eta}_Q$. This is an illustrative sufficient condition, not an experimentally tested gate.

The expression $B(I - D)^{-1}C$ enumerates visual-to-state-to-visual paths in the nonnegative comparison system, including state recurrence through powers of D. The term $B(I - D)^{-1}\boldsymbol{\eta}_Q$ carries state-output residuals into later visual calls. Feedback removes these learned state-error paths from the bound, but visual history, actions, and supplied-state error remain. An observed improvement therefore requires empirical evaluation; it cannot be deduced from the presence of a removed path.

A.3 Counterexample: exact state replacement can increase realized error

Let the true sequence satisfy $v_j^* = q_j^* = 0$, with exact initial state $v_0 = q_0 = 0$. Consider the frozen predictor

$$\widehat{v}_{j+1} = \widehat{q}_j + 1, \qquad \widehat{q}_{j+1} = -1.$$

At the first call both rollouts predict $\widehat{v}_1 = 1$. Free rollout then stores $\widehat{q}_1 = -1$, yielding $\widehat{v}_2 = 0$. Feedback stores the exact value $q_1 = 0$, yielding $\widehat{v}_2 = 1$. In two-step notation, $A = C = D = 0$,

$B_{21} = 1$, and both residual vectors equal $(1, 1)^\top$. The common free upper bound is $(1, 2)^\top$, and the feedback upper bound is $(1, 1)^\top$. Nevertheless, realized errors are $(1, 0)^\top$ and $(1, 1)^\top$, respectively. The state error canceled a visual prediction bias before replacement. This analytic construction proves the limitation of an ordering argument; no such cancellation is identified by the reported experiments.

A.4 Proof of Lemma 2 and pairwise margins

Let $U^* \in \arg\min J$ on the same finite bank. By the assumed optimization gap,

$$J(\widehat{U}) - J(U^*) = \widehat{J}(\widehat{U}) - \widehat{J}(U^*) + \epsilon(U^*) - \epsilon(\widehat{U}) \tag{14}$$

$$\leq \eta_{\text{opt}} + \sup_{U \in \mathcal{U}} \epsilon(U) - \inf_{U \in \mathcal{U}} \epsilon(U). \tag{15}$$

If $|\epsilon| \leq \delta$, the familiar weaker bound $\eta_{\text{opt}} + 2\delta$ follows. The oscillation form makes invariance to a common additive cost bias explicit. Exact argmin on a saved bank has zero optimization gap. A final CEM mean is not necessarily even a member of that bank, and its optimization gap is not certified; the claim must not be applied to it without a defined comparison set and a valid gap.

For an actual-cost-preferred reference w and searched sequence p, define $m = J(p) - J(w) > 0$. Then

$$\widehat{J}(w) - \widehat{J}(p) = -m + \epsilon(w) - \epsilon(p). \tag{16}$$

A strict predicted reversal occurs exactly when $\epsilon(w) - \epsilon(p) > m$. A numerical reversal tolerance τ changes this to $m + \tau$. Common bias cancels. These conditions do not imply physical improvement because actual cost and physical error can rank actions differently.

A.5 Prediction MSE, goal residuals, and cost disagreement

Cost error under squared feature distance.

For $c(y, y_g) = \|y - y_g\|^2/D$, put $r = y - y_g$ and $d = \widehat{y} - y$. Expanding the square gives

$$\epsilon = \frac{2r^\top d + \|d\|^2}{D}, \qquad |\epsilon| \leq \frac{2\|r\|\|d\| + \|d\|^2}{D}. \tag{17}$$

With scalar $y = 1, y_g = 0$, errors $d = 1/2$ and $d = -1/2$ have equal MSE $1/4$ but cost errors $5/4$ and $-3/4$. These are illustrative values, not measurements. For DINO-WM, apply the expansion separately to visual and state terms with their original dimensions and weights. Replacing this by one unweighted concatenated average would change the objective. For supplied terminal state, its approximation error also contributes to total cost error; a visual-only bound cannot silently remove it. V-JEPA2's offline L1 metric is not reinterpreted as squared distance.

Objective disagreement persists under exact prediction. If physical errors satisfy $e(w) < e(p)$ but actual costs satisfy $J(w) > J(p)$, exact prediction $\widehat{J} = J$ preserves that physically undesirable comparison. This shows a limitation of that goal cost on that pair, not a lack of information in the representation. A different readout could use the same features differently. Conversely, an actual-cost-aligned pair permits a predictor-reversal diagnosis without proving objective alignment elsewhere.

A.6 State replacement as a restricted affine projection

For a fixed action sequence and step, the known-state constraint set is $\mathcal{C}_U = \{(V, Q) : Q = \mathcal{B}\bar{q}(U)\}$. Under the Euclidean norm on the two blocks, the equality-constrained least-squares projection (Boyd & Vandenberghe, 2018, Chapter 16), i.e., the unique closest point to $(\widetilde{V}, \widetilde{Q})$ is

$$\Pi_{\mathcal{C}_U}(\widetilde{V}, \widetilde{Q}) = (\widetilde{V}, \mathcal{B}\bar{q}(U)).$$

Minimizing the separable squared distance sets $V = \widetilde{V}$; the state constraint fixes Q. Applying this map twice changes nothing. This is a projection onto a constraint on one supplied substate, not onto the full set of physically valid visual states. If the supplied state is wrong, the true feature need not lie in this set. Even exact projection does not constrain the next visual prediction's error or its physical consequence.

For n_p patch-state vectors with mean $\mu_q = n_p^{-1} \sum_i q_i$, the bias–variance identity is

$$\sum_{i=1}^{n_p} \|q_i - q^*\|^2 = \sum_{i=1}^{n_p} \|q_i - \mu_q\|^2 + n_p \|\mu_q - q^*\|^2.$$

Patch agreement alone does not establish a correct shared value. Mean broadcasting is the orthogonal projection onto the equal-patch subspace. Our Spatial control instead broadcasts the first predicted patch, as implemented before this analysis; it is not that mean projection. It removes patch disagreement while retaining a learned state value that can continue to carry errors into later predictions.

A.7 Fixed-sequence controls and common actuator finishing

At a fixed terminal action sequence, Endpoint, Spatial, and Feedback all use the same propagated terminal state cost. Subtraction cancels it exactly, proving Equation 5. This comparison isolates differences in their visual costs on that sequence. Different selected actions generally have different propagated terminal states, so the cancellation does not establish mediation of the physical gains of reoptimized plans.

For one spatial axis, the ten internal PD substeps use a substep matrix

$$S = \begin{bmatrix} 1 - k_p \Delta t^2 & \Delta t(1 - k_d \Delta t) \\ -k_p \Delta t & 1 - k_d \Delta t \end{bmatrix}, \quad P = S^{10}, \quad A_c = \begin{bmatrix} 1 & P_{01} \\ 0 & P_{11} \end{bmatrix}, \quad b_c = \begin{bmatrix} 100(1 - P_{00}) \\ -100 P_{10} \end{bmatrix}.$$

Matrix entries P_{01}, P_{11} use zero-based indices. Here $\Delta t = .01$; $k_p = 100, k_d = 20$ yield the exact-gain illustration $S = [.99, .008; -1, .8]$. The early RGB-localized experiments use these exact gains. The later RGB-calibrated cohort and the generated DINO-WM/LeWM common-finish studies use the frozen fitted gains $k_p = 102.1442, k_d = 20.4151$, with full precision retained in the configuration. Table 9 distinguishes their observation and action conventions. The command target is a displacement from the current pusher position. For $s = (p, v)^\top$, unrolling $s_{t+1} = A_c s_t + b_c u_t$ gives $s_T = A_c^T s_0 + \mathcal{C}_T U$. When $\mathcal{C}_T$ has full row rank, the standard minimum-energy endpoint solution (Boyd & Vandenberghe, 2018, Chapters 16–17) is

$$U = \mathcal{C}_T^\top (\mathcal{C}_T \mathcal{C}_T^\top)^{-1} (s^* - A_c^T s_0). \tag{18}$$

The Lagrangian of minimizing $\frac{1}{2}\|U\|^2$ subject to the endpoint equality has stationary point $U = -\mathcal{C}_T^\top \lambda$, with $\lambda = -(\mathcal{C}_T \mathcal{C}_T^\top)^{-1}(s^* - A_c^T s_0)$, proving the expression. Strict convexity gives uniqueness. An unconstrained solution does not prove compliance with action limits. The original bounded implementation checks feasibility rather than silently clipping; the generated DINO-WM and LeWM common-finish runners record clipping safeguards at $[-1, 1]$, inactive in all completed finishing trajectories.

The subsystem contains no object contact dynamics. Common finishing can move the object, and different first-stage plans leave different contact configurations. Therefore the protocol tests object benefit after final actuator precision is controlled, not an invariant contact history or an isolated contact-prediction mechanism. Calibrated dynamics, finite history, and the native cost normalization remain part of the experimental conditions.

B EXPERIMENT PRINCIPLES AND COVERAGE

Each panel in Figures 5–8 specifies the cohort, intervention, and endpoint for one completed experimental family. Axes denote command indices except where labeled. Circles mark observations, diamonds endpoints; bars show allocated computation or execution, not measured trajectories. The additional common-bank and scoring-time controls have separate schematics below.

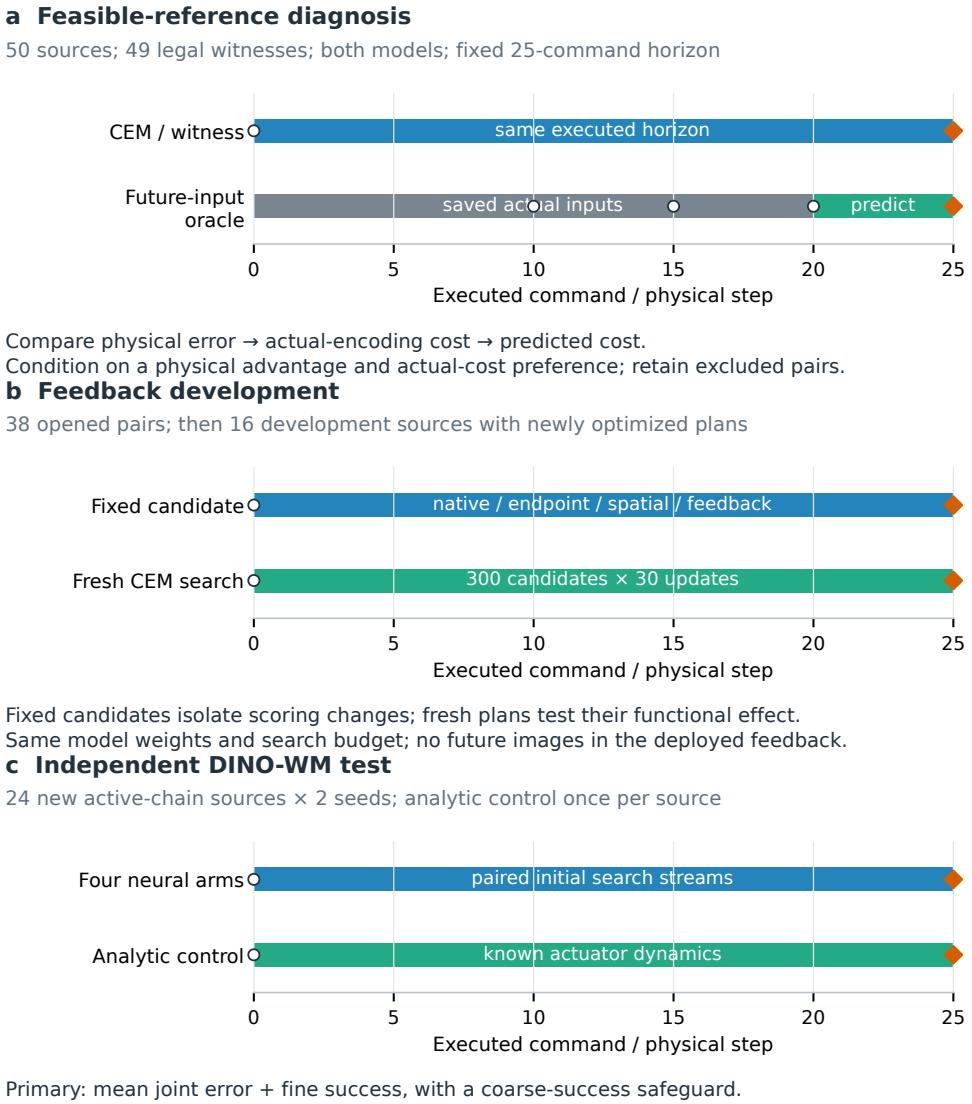


Figure 5: Diagnostic and initial intervention designs: (a) execute searched and feasible-reference sequences over the same horizon, then compare predicted and actual costs; (b) score fixed pairs before searching new plans on development sources; (c) compare four frozen-model variants and analytic control on 24 independent active-chain sources. Future images in the oracle are evaluation-only.

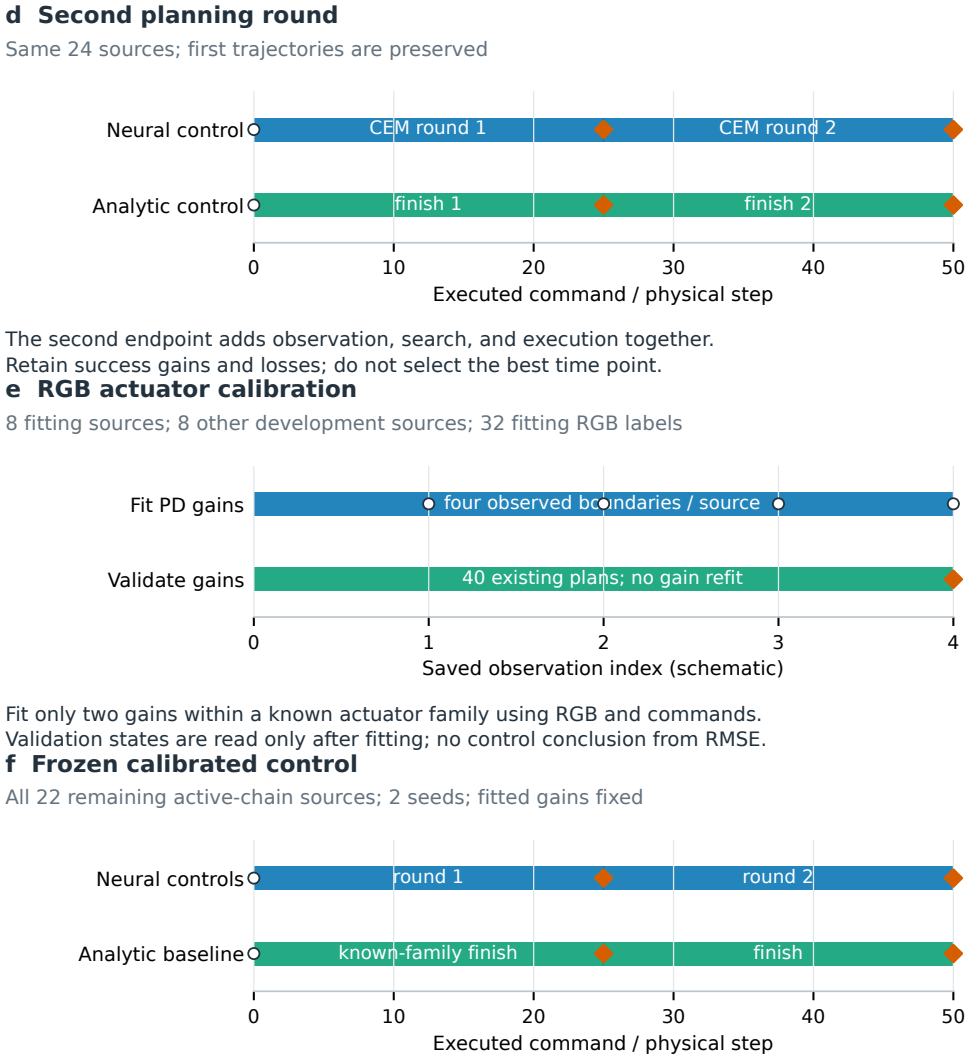


Figure 6: Extension and calibration designs: (d) add one planning round to the same trajectories, retaining both endpoints; (e) fit two actuator gains on eight development sources and validate on eight other development sources; (f) freeze the fitted gains before evaluating all 22 remaining active-chain sources. Different cohorts do not identify the causal effect of replacing exact gains.

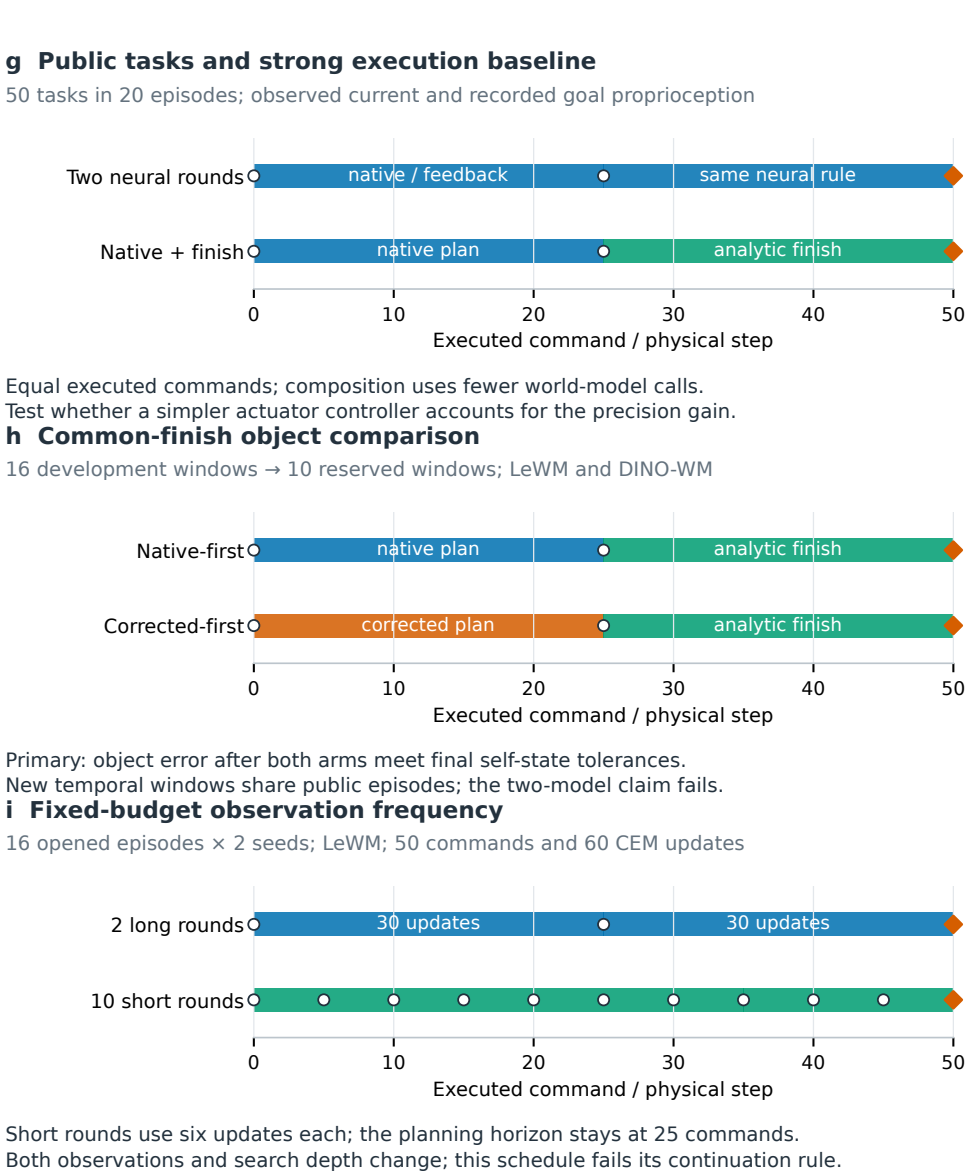


Figure 7: Functional baseline designs: (g) compare two neural rounds with a neural plan followed by an analytic finish on 50 public tasks; (h) compare native-first and corrected-first plans under the same finishing rule, first in 16 development windows and then in ten reserved windows, for both models; (i) redistribute a fixed 50-command and 60-update budget across observation rounds. The last intervention changes both observation frequency and search depth.

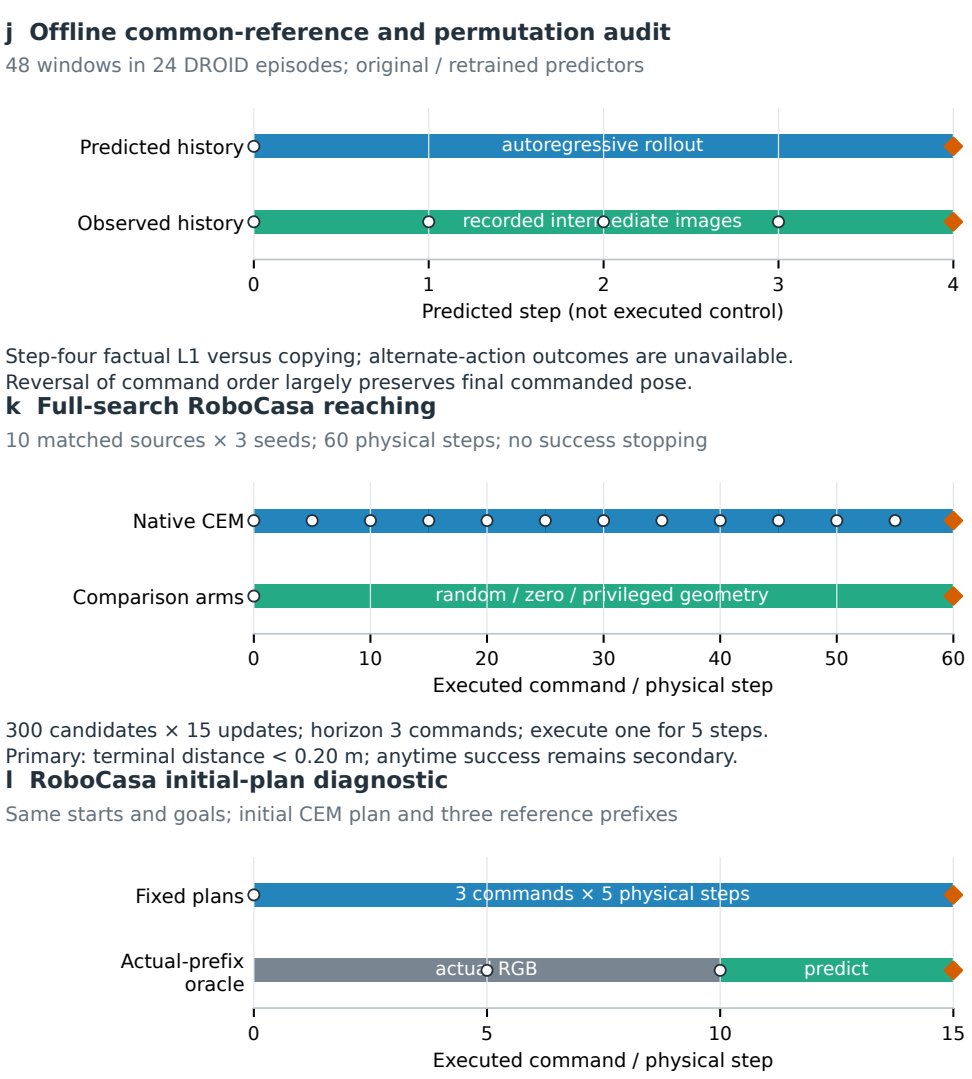


Figure 8: V-JEPA2 designs: (j) compare original and retrained predictors on the same offline windows and audit the meaning of action permutations; (k) run full-search native RoboCasa reaching with fixed goals and comparison controllers for 60 steps; (l) execute each initial plan and three reference prefixes separately for 15 steps, including the actual-prefix oracle. The offline metric, conditional 15-step diagnostic, and 60-step physical endpoint remain distinct. Figure 33 separately shows the completed scoring-time controls.

C RECORDED SIMULATION IMAGES AND SELECTION RULES

Image panels in this appendix use RGB arrays saved in the PushT and RoboCasa environments (Chi et al., 2024; Nasiriany et al., 2024). We make no learned-pixel reconstructions: these latent models do not supply a pixel decoder for the displayed rollouts. Each exported array is accompanied by its source HDF5 hash, dataset path convention, source identifier, seed, and command index. Image content is not synthesized or retouched. Figure layouts rescale images with their original aspect ratios and place annotations outside the image regions.

The recorded PushT example uses stored row zero and seed index zero, selected by file order, with all four methods shown at command 25. The supplement includes terminal images for every one of the 24 sources at that same seed; the quantitative results retain both seeds. The RoboCasa sequence panels are descriptive post hoc examples: within terminal success, reached-then-left, and never-reached categories, choose the first source and seed in numerical order. This displays each outcome category without claiming its selected example is typical. All 30 native valid-goal terminal frames are additionally shown. No example replaces an aggregate endpoint or supports a frequency estimate.

The RoboCasa camera view may make small hand–object distance changes difficult to see. Physical distance is recomputed from saved simulator coordinates; a label below a frame refers to that frame's recorded step. Crossing 0.2 m is the reaching predicate used here, not evidence of grasping or completion of the underlying kitchen task. The supplied slide-show animations display saved observation boundaries at an explanatory playback rate; they are not continuous real-time recordings.

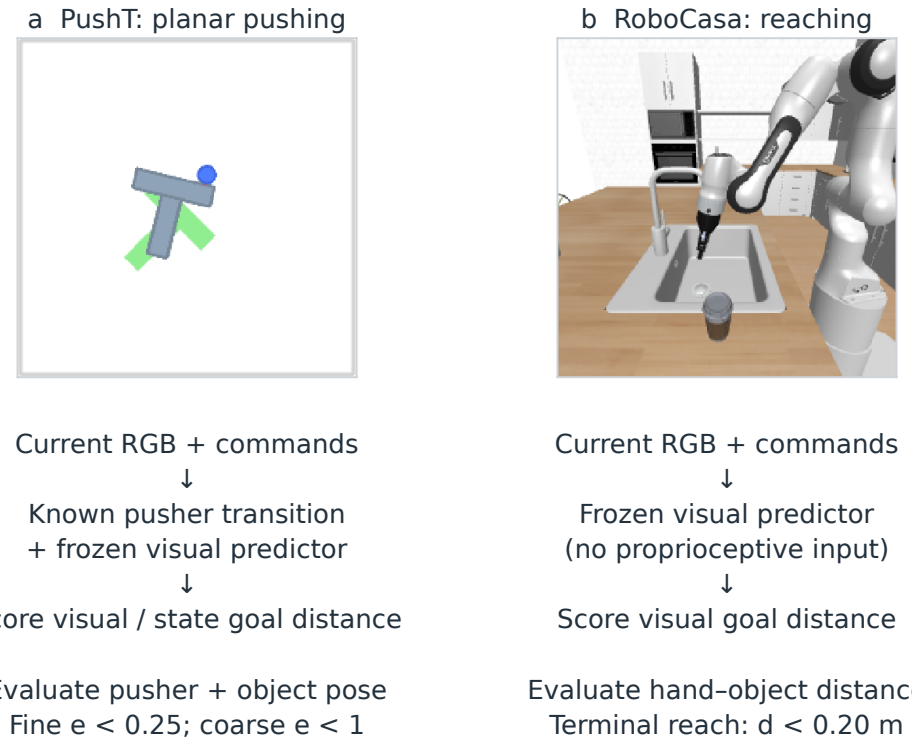


Figure 9: Simulation and evaluation interfaces. Actual saved images illustrate planar PushT and RoboCasa reaching. The schematic separates known actuator state, unknown object dynamics, image goal cost, and physical evaluation. Positions and distances are evaluated in simulator coordinates; image appearance alone does not define success. The RoboCasa reference may use privileged geometry, but the native planner does not receive it.



Figure 10: Recorded PushT terminal frames, sources 0–5 of 24, at the first optimizer seed. Fixed column order: goal, native, endpoint, spatial, feedback; labels give joint endpoint error. Every source is included across these four pages; quantitative analyses use both seeds.



Figure 11: Recorded PushT terminal frames, sources 6–11 of 24, at the first optimizer seed. Fixed column order: goal, native, endpoint, spatial, feedback; labels give joint endpoint error. Every source is included across these four pages; quantitative analyses use both seeds.



Figure 12: Recorded PushT terminal frames, sources 12–17 of 24, at the first optimizer seed. Fixed column order: goal, native, endpoint, spatial, feedback; labels give joint endpoint error. Every source is included across these four pages; quantitative analyses use both seeds.



Figure 13: Recorded PushT terminal frames, sources 18–23 of 24, at the first optimizer seed. Fixed column order: goal, native, endpoint, spatial, feedback; labels give joint endpoint error. Every source is included across these four pages; quantitative analyses use both seeds.

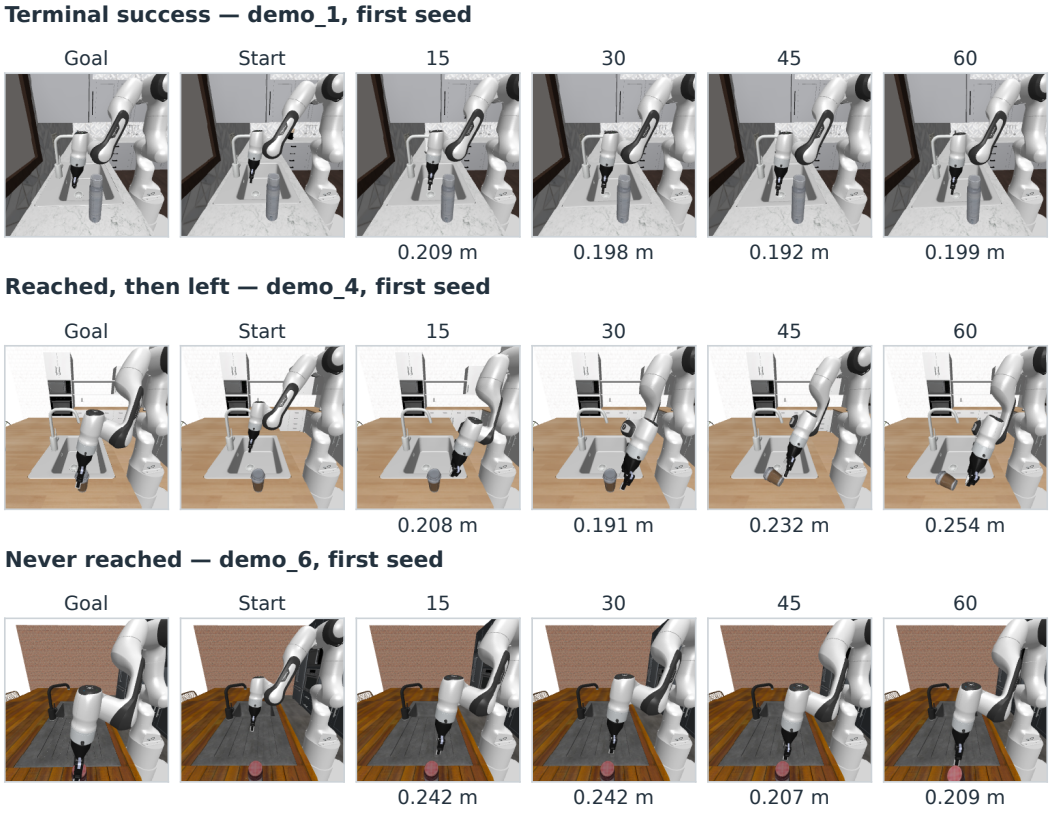


Figure 14: Saved RoboCasa observations for the first numerical source/seed in each outcome category. Columns show the fixed goal, start, and steps 15, 30, 45, and 60. Below-image distances use the same saved states. Reached-then-left is determined from the full 60-step trace, so its minimum can lie between displayed frames. Selection is post hoc and descriptive; no stopping policy is evaluated.

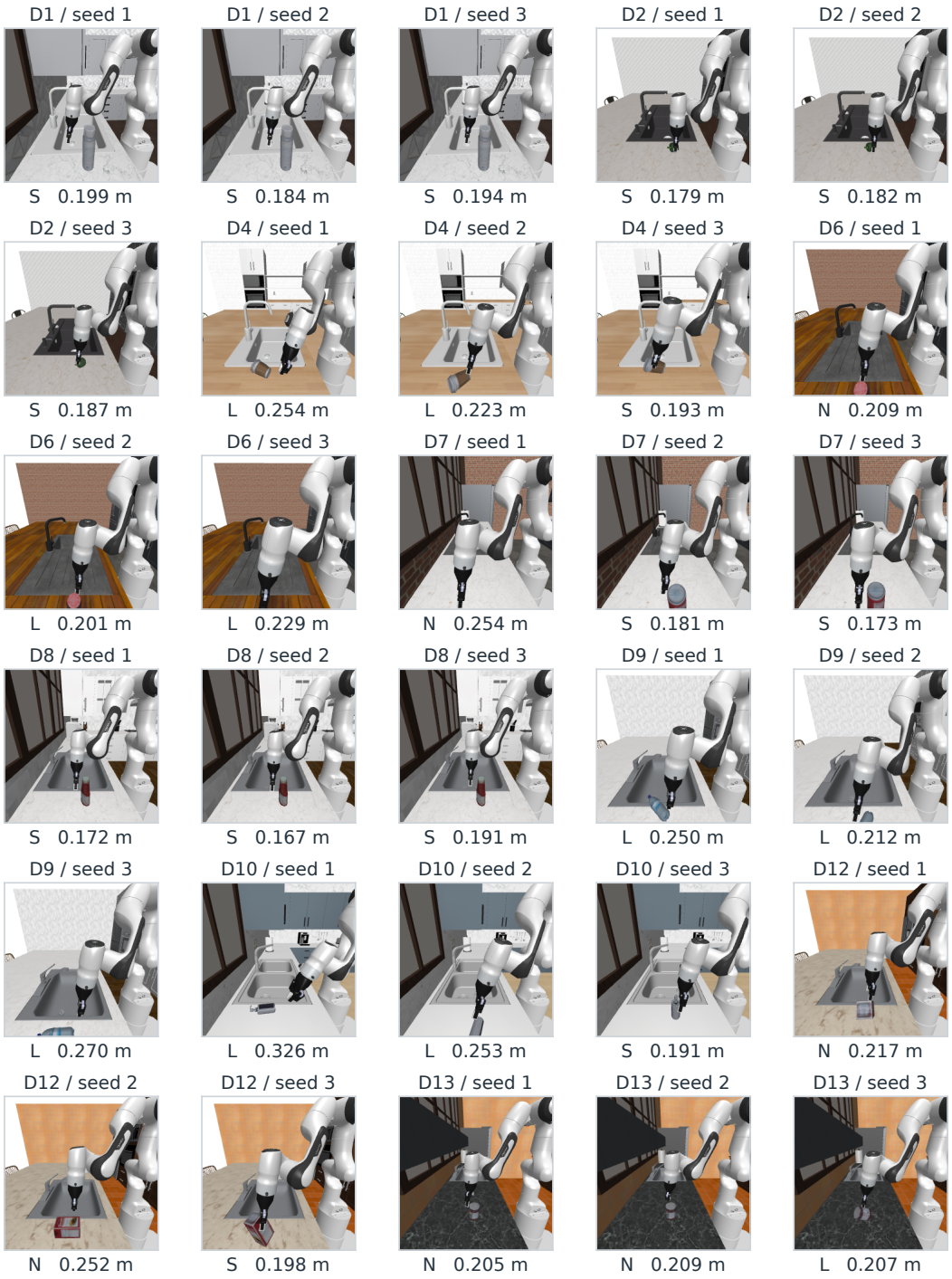


Figure 15: All 30 native valid-goal terminal images, in source/seed order, with terminal hand–object distance. S denotes terminal success, L arrival followed by terminal failure, and N no arrival. Category counts are 14, 10, and 6. These images come from the completed full-search experiment, not the smaller earlier pilot.

Table 3: Nearby contributions and the narrower question studied here. This is a conceptual comparison, not an empirical ranking against these methods.

Work	Relevant contribution	Distinction in this study
DINO-WM / LeWM (Zhou et al., 2025; Maes et al., 2026)	Latent visual prediction for goal planning	Audit preferences using actual executed outcomes at the same horizon.
V-JEPA2 / JEPA-WMs (Assran et al., 2025; Terver et al., 2026)	Action-conditioned video prediction and systematic planning design	Use the retrained native interface; do not claim its self-state feedback benefit.
RAC (Hu et al., 2022)	Separate robot/world dynamics and visual costs	Retrofit a frozen checkpoint and control endpoint scoring and broadcasting.
ACID (Seo et al., 2026)	Inverse-action consistency for planning	Measure reversals against actual outcomes; no inverse model is trained.
PSG-JEPA / TC-WM (Yan et al., 2026; Fu et al., 2026)	State-grounded or task-centric representation learning	Hold model weights fixed; no claim that the representation lacks information.
LeWMRO (Vesessook & Yang, 2026)	Scored-horizon and executed-prefix mismatch	Retain scoring-time explanations; no novelty claim for prefix/running costs.
Traj-LeWM (Huang et al., 2026)	Learned trajectory-level planning cost	Study the deployed native cost without training a new trajectory evaluator.
Decision-focused learning (Mandi et al., 2022)	Feasible-decision ranking as a learning objective	Use paired costs for diagnosis; do not propose a new ranking-loss theory.

D SCOPE RELATIVE TO ADJACENT WORK

Our experiments compare the explicit implemented controls; this table should not be read as evidence of superiority over a method that was not run.

E LeWM fixed-bank and common-finish boundary tests

E.1 Cohorts and evidence status

The fixed-bank test uses generated-task IDs 0–9; the functional test uses IDs 0–15. These were the predetermined leading IDs from the same 32-task generator used in the DINO-WM common-finish comparison, without selecting tasks by LeWM outcome. They are new LeWM evaluations on an already observed task distribution, not new environments or certified pretraining OOD. The two LeWM tests overlap in sources. The first ten and first sixteen were fixed before inspecting their LeWM results.

We use the LeWM model and planning interface (Maes et al., 2026) with the PushT simulator distributed with Diffusion Policy (Chi et al., 2024). The simulator generates each source and a legal goal-producing sequence before planning. Task construction, initial states, goal observations, actuator calibration, model/scaler identities, and both optimizer seeds are saved. The existing frozen model has a 192-dimensional visual feature interface; it does not expose DINO-WM's patch-level proprioceptive block for replacement. The functional intervention is therefore the previously implemented terminal constraint, not an additional feedback architecture.

E.2 P0: all fixed candidates, features, and decisions

Each bank contains zero action and 15 random sequences sampled from the existing initial CEM standard-normal action distribution, transformed by its scaler and clipped to the legal interval. No CEM optimization is performed. Candidates are sampled once from the existing initial proposal distribution. All 160 sequences are committed before the first score or candidate execution. Every sequence executes 25 commands from its saved initial state. Scoring retains the original squared-distance cost, the same frozen encoder for actual and predicted outcomes, and the same goal and horizon. All 1,200 unordered pairs are retained, including ties and reversals; statistical inference uses ten tasks rather than candidates or pairs.

Table 4: LeWM P0, all 16 candidates per task. Intervals resample ten tasks. Physical gaps use object error at command 25. Bootstrap intervals describe across-task variation. Original sign-flip values and the assumptions needed to interpret them are discussed below.

Quantity	Mean	95% interval
Normalized actual-cost regret	.06530	[.00000,.15577]
Uniform normalized regret	.61456	[.55086,.66644]
Prediction minus uniform	-.54926	[-.64369,-.44234]
Raw actual-cost regret	12.26867	[.00000,32.10670]
Total physical gap	.07436	[.01585,.14653]
Signed prediction difference	-.00308	[-.08259,.06661]
Actual-cost selection gap	.07744	[.01057,.15122]

Seven tasks have zero actual-cost regret. The other three normalized values are .43473, .18294, and .03528. Removing the zero-action candidate leaves the same selected candidates and regret, with uniform normalized regret .61910; this is an overlapping subset, not an independent repeat. Zero action is never selected by the predicted, actual-cost, or physical criterion. The actual-cost and physical argmins agree on six tasks. The 136 strict predicted/actual pair reversals and 156 actual-cost/physical reversals describe local orderings; they do not replace the top-choice endpoint or change the task-level denominator.

Inference for nonnegative bank gaps. The mean over these executed banks is a known descriptive quantity; its percentile bootstrap interval summarizes variation under resampling the sampled tasks (Efron, 1979). Seven of ten regret values are zero. The probability that an ordinary ten-draw bootstrap sample contains only zeros is $(7/10)^{10} = .0282475$, exceeding the lower 2.5% quantile. A zero percentile lower bound is therefore expected. It does not negate the observed mean of .06530.

Four tasks have a positive actual-cost physical selection gap and six have zero. In a two-sided mean sign-flip calculation, the smallest attainable value for four nonzero, same-sign differences is $2/2^4 = .125$; flipping zero entries adds no distinct statistic. Enumerating all 2^{10} signs is compu-

tationally exact, but inferential exactness requires an appropriate sign-exchangeable or symmetric null (Hemerik & Goeman, 2018). Definitionally nonnegative regret and physical gaps do not, by themselves, justify such a null as an assumption-free test of a positive population mean. We retain the original sign-flip results, observed means, and bootstrap summaries; they answer differently conditioned questions and are not conflicting numerical audits. The total physical gap likewise has four nonzero task values and the original value .125.

For signed paired controller or decision-rule differences, we retain the frozen procedure, conditional on sign exchangeability under the null. LeWM and RoboCasa enumerate all signs for 16 and ten tasks, respectively; the 32-task DINO-WM analyses use 50,000 Monte Carlo sign flips with the recorded plus-one correction (Hemerik & Goeman, 2018). Seed means precede task resampling, and intervals remain pointwise. No test is substituted to obtain a more favorable result.

The raw records include predicted, actual, and goal features, all candidate actions, terminal states, saved RGB, and full execution trajectories. Independent float64 cost recomputation differs from saved device costs by at most 4.24×10^{-5}, without changing any candidate selection. This check establishes argmin stability, not stability of every pairwise reversal classification under the precision change. Full state and RGB replay checks pass for all candidates and sources. Validation commands are accounted for separately from the 4,000 scientific execution commands.

E.3 P1: NATIVE VERSUS TERMINAL CONSTRAINT AFTER COMMON FINISHING

Both arms use 300 candidates, 30 elites, 30 CEM updates, and 25 first-stage commands, followed by the same 25-command analytic finish. The two base seeds are 2026092462 and 2026092463, with the saved source-specific offsets. The terminal constraint uses the existing position/velocity scales of 20 and a fixed first-iteration median-based weight. Frozen calibration, action limits, sources, and stopping rules are unchanged. There are 64 retained trajectories and no tolerance-based exclusions. The finish's action-clipping safeguard is inactive in every completed trajectory.

Table 5: LeWM P1 absolute and paired outcomes. The primary comparison is command-50 object error, with two seeds averaged within each of 16 tasks. Fine and coarse counts are executions.

Command-50 quantity	Native	Constraint	Constraint minus Native
Mean object error	.72304	.70058	-.02246 [-.12326,.09080]
Fine successes	11/32	13/32	+6.25 pp [-6.25,18.75]
Coarse successes	26/32	26/32	0 pp
Max pusher error (pixels)	.19057	.01119	—
Max velocity error (pixels/s)	.04313	.02270	—

The primary exact sign-flip test gives $p = .70367$. Fine success gains four executions and loses two ($p = .625$); all coarse classifications are identical, which is not a population noninferiority result. Both seed-wise object-error point estimates are lower under the constraint, but the task-level interval still spans benefit and harm. Position and angle component intervals also cross zero. At command 25, the paired joint-error change is -.30079 [-.46409,-.12175] ($p = .00601$), while the object-only change is -.06722 [-.20005,.08578] ($p = .38626$). Better initial actuator precision therefore does not supply the missing post-finish object evidence.

Mean object displacement during finishing is 6.05865 pixels for Native and 2.57378 for Constraint. This descriptive stage difference is not a mediation estimate. Every final pusher is within the prespecified 1-pixel and 1-pixel/s tolerance. This remains a descriptive difference between finishing trajectories, not an explanation of the primary effect.

Earlier ten reserved windows yielded -.24158 [-.51402,-.02786] under LeWM and +.05825 [-.01104,.15513] under DINO-WM. Those windows share source episodes and differ from the generated-task cohort. They are not pooled with P1. The change in significance does not establish an interaction between cohort and intervention. The model/interface contrast likewise does not identify an architectural explanation for the difference between the two interventions.

E.4 Engineering record and reproducibility map

The original failure is retained alongside the corrected dispatch. No scientific configuration, sample, or threshold was changed. Run entries below are relative to `evidence/results/DIRECT_VISUAL_CONTROL_20260912/`.

Table 6: LeWM execution record. CPU audits of saved data add no model or dynamics calls.

Stage	Model / dynamics work	Configuration change	Exit	Record
P0 original	None; path error before import	None	2	P0 original
P0 recovery	160 candidates plus replay	Path fix only	0	P0 recovery
P1	64 plans plus replay	None	0	P1
CPU audit	Saved records only	None	0	Analysis

The log entries are `reviewer_lewm_p0_20260925/execution.json`, `reviewer_lewm_p0_recovery1_20260925/execution.json`, and `reviewer_lewm_p1_20260925/execution.json`; CPU checks are emitted by the analysis entry below. The recovery used the remaining 899 seconds of the original 900-second budget and finished in 48.18 seconds; P1 used 69.61 of 1,800 seconds. These timing details establish execution status, not scientific support.

Local audits read all 160 P0 trajectories and all 64 P1 trajectories, including the 1,920 P1 CEM updates. Saved states, RGB, decisions, feature costs, finish actions, task identities, and exit records were checked. Local and remote statistics differ in 15 of 1,174 numeric entries only at floating-point rounding, with maximum absolute difference 1.78×10^{-15}; selections, intervals at reported precision, tests, and conclusions are unchanged. Independent HDF5 endpoint recomputation matches the primary result within 10^{-12}. These are numerical and replay audits of the same data, not independent scientific replications.

The anonymous package stores these stages under `evidence/results/DIRECT_VISUAL_CONTROL_20260912/`: `reviewer_lewm_p0_20260925` retains the initial failure; `reviewer_lewm_p0_recovery1_20260925` contains the complete fixed bank; and `reviewer_lewm_p1_20260925` contains both functional arms. The entry `analysis/reanalyze_lewm_generality.py` recomputes statistics and audits the saved records. Source implementations are under `experimental_source/reviewer_lewm_generality_20260925/`. Main figure generators read the saved inputs directly. The entry `analysis/verify_finite_history.py` numerically checks the matrix identities and counterexample; it does not substitute for the proofs.

F Extended protocols and complete results

F.1 How to read the supplementary evidence

The experiments answer different questions and retain their original decisions. The diagnostic conditions on executable reference comparisons; the intervention pilot is development; the 24-source experiment tests independent active-chain sources; later composition studies test temporal windows in shared public episodes. Repeated optimizer seeds are not independent sources. None of these source splits establishes a pretraining holdout. Negative complete criteria are retained alongside positive component results.

The following algorithm and protocol sections support implementation. Subsequent sections retain actuator calibration, complete outcome tables, failed criteria, component plots, and the V-JEPA2 contrast audit. Completed RoboCasa outcomes, the fixed-bank decision audit, generated-task common finishing, goal-semantic control, and scoring-time controls are all included. The latter controls retain their opened-source development scope.

Appendix E contains the completed new LeWM bank and common-finish comparison. These share generated sources with DINO-WM but have a different state interface and intervention. The main text separates this correction-interface boundary from the earlier positive temporal-window result.

Table 7: Evidence inventory. Counts describe statistical units; repeated seeds are shown separately.

Study	Scope and role	Units
Feasible references	Two-model conditional diagnostic	50 groups
Feedback pilot	Opened comparisons and new plans	38 pairs; 16 groups
DINO-WM confirmation	Independent active-chain sources	24 groups $\times$ 2 seeds
Second planning round	Same sources; longer execution	24 groups $\times$ 2 seeds
RGB-calibrated control	Separate remaining sources	22 groups $\times$ 2 seeds
Public task transfer	Opened task selection; full goal state	50 tasks; 20 episodes
Two-model composition	New windows in shared episodes	16 episodes $\times$ 2 seeds
Corrected-first validation	Reserved temporal windows	10 episodes $\times$ 2 seeds
V-JEPA2 offline	Common-reference development	48 windows; 24 episodes
RoboCasa full search	Completed diagnostic and native control	96 trials; 10 matched sources
Common candidate audit	Post hoc analysis of fixed banks	10 banks; 160 candidates
Generated common finish	Four DINO-WM arms; Spatial added on opened cohort	32 tasks $\times$ 2 seeds
LeWM fixed bank	Same generator; no outcome-based candidate filter	10 tasks; 160 candidates
LeWM common finish	Native versus terminal constraint; benefit unconfirmed	16 tasks $\times$ 2 seeds
Goal semantics	Controlled translation; measurement repair disclosed	10 sources; 70 candidates
Scoring time	Same first seed; original source cohort	10 paired sources per arm

F.2 Algorithm and observation interfaces

Algorithm 1: score a DINO-WM candidate with known-state feedback. This algorithm covers the overwrite used in both the earlier RGB-localized studies and the generated common-finish experiment; initialization, actuator gains, and action preprocessing follow the distinct columns of Table 9. The baseline frozen rollout and visual features come from DINO-WM and DINOv2 (Zhou et al., 2025; Oquab et al., 2024); the replacement and matched controls are our interventions. Inputs are a current observation, available self-state, a legal candidate sequence, a goal interface, frozen visual predictor and state encoder, and a known actuator transition. No executed future, object state, or physical success signal is a scoring input.

1. Apply the experiment's action preprocessing once. Feed exactly the same raw commands to actuator propagation and their checkpoint-normalized chunks to the predictor. Initialize latent history with the observed encoding.
2. Propagate the known actuator through the candidate's raw commands to obtain the state at each five-command boundary.
3. For each chunk, call the frozen predictor on the permitted latent history and candidate action chunk. Retain its predicted visual channels.
4. For feedback, replace the new proprioceptive channels with the existing encoder's embedding of the known propagated state, broadcast over patches. For the spatial control, instead broadcast the first patch's predicted state at intermediate boundaries. For native and endpoint-only control, keep intermediate outputs unchanged.
5. Append the resulting latent and continue prediction with the next action chunk. At the terminal boundary, every correction uses the same known state for its proprioceptive cost. Native uses its prediction.
6. Return visual goal MSE plus proprioceptive goal MSE, using each term's original normalization. Physical joint error is never optimized directly.

Outer search. We use the cross-entropy method (CEM) (de Boer et al., 2005); initialize proposals in the checkpoint's normalized action coordinates. At each iteration score 300 candidates, retain 30 elites, and update their mean and sample standard deviation. The mean is inserted as the first candidate. After 30 iterations execute the final mean under that experiment's action convention, rather than the best historical sample. Original RGB-localized studies clip raw commands to $[-1, 1]$; the public DINO-WM variant retains its native proposal convention. No success-based early stopping is used. Full transition equations and normalization details appear below.

LeWM adapter. Add a propagated terminal self-state penalty to LeWM's native visual cost (Maes et al., 2026); do not invent a writable state channel. The coefficient is determined from the initial candidate batch and frozen within that planning call. The exact penalty is given in the LeWM section below.

Execution composition. Execute the first 25 commands, observe the resulting current self-state, and solve a new 25-command analytic finishing problem. Compare native-first and corrected-first composition using the same finishing rule and total commands. Their intermediate states and finishing actions can differ. The comparison isolates neither an identical contact trajectory nor identical computational work.

Table 8: Observation and goal interfaces must not be pooled. Object state is used only for evaluation, except in the explicitly privileged RoboCasa reference.

Setting	Current and goal self-state	Information supplied during planning
RGB-localized DINO-WM	Position from RGB; current velocity from past commands; goal velocity zero	Current image, candidate commands, known actuator model
Public PushT variants	Observed current and recorded goal proprioception	Same interface for all compared arms; no object state
V-JEPA2 offline	Shared recorded feature/action cache	Two/four-frame context; observed history is an oracle
RoboCasa native	Current RGB and fixed goal RGB	No proprio input; frozen retrained predictor

Table 9: Actuator and observation conventions. Early RGB-localized studies and the generated common-finish experiments use distinct input protocols. Numerical gain values are rounded here; saved calibration files retain full precision.

Item	Earlier RGB-localized protocol	Generated common-finish protocol
Initial self-state	Position localized from current RGB; velocity propagated from past executed commands	Observed current pusher position and velocity
Goal self-state	RGB-localized position; zero target velocity	Recorded goal position and velocity, preserved
Actuator gains	Original studies: $100/20$; later 22-source study: frozen RGB calibration	Frozen calibration $102.1442/20.4151$ in both models
Search commands	Normalized CEM proposals mapped to raw commands and clipped to $[-1, 1]$	DINO-WM public proposal convention, without raw clipping; LeWM scaler and $[-1, 1]$ clipping
Analytic endpoint actions	Original bounded analytic controller checks feasibility; not the same two-stage protocol	Minimum-energy finish from each arm's own observed state; clipping safeguard at $[-1, 1]$, zero activations in completed runs
Goal-cost interface	Visual MSE plus state MSE with original dimensions	DINO-WM retains the same two-term normalization; LeWM retains visual cost plus its separate terminal penalty

F.3 RoboCasa full-search protocol and completed results

This fixed experiment completed all 96 trials with exit code zero in 88,873 seconds. The CPU audit checked 1,275 execution chunks, 492 full CEM plans, 5,760 control steps, and 615 diagnostic steps; maximum distance-recomputation error was 1.11×10^{-16} m and maximum feature-cost discrepancy was 1.06×10^{-7}. It is not a third-model method success or an independent-scene validation. The retrained no-proprioceptive V-JEPA2-AC configuration (Assran et al., 2025; Terver et al., 2026) does not expose the DINO-WM state-replacement interface. The experiment therefore tests native control and the conditional action-comparison diagnostic.

The source environment contains 14 demonstrations, of which 11 have the required reach segment. The source segment endpoint does not itself meet the reach predicate in these 11 cases; ten have an

immediately following qualifying frame. The remaining case has no valid paired goal and remains in the original-goal analysis. All eligible sources and this selection rule were fixed before this run. They were already opened during development.

The native controller uses 300 CEM samples, 15 iterations, 10 elites, a three-command horizon, two-frame context, one command repeated for five physical steps, and 60 physical steps without early stopping. The primary endpoint is terminal hand–object distance below 0.2 m. Anytime success is secondary. Ten valid-goal sources use three optimizer seeds; eleven original-goal sources use the first seed. Zero and privileged geometry references run once per source; random control uses all three seeds. The primary comparison is valid-goal native versus random on the same ten sources, averaging seeds within source before 10,000 paired bootstrap draws. The paired original-goal ablation uses only the common sources and first seed. A two-sided source-level sign-flip test is also reported, conditional on the sign-exchangeability assumption.

For each of the 41 native trials, its fixed first full three-command plan is executed separately for 15 steps. Zero, same-seed random, and privileged-geometry reference prefixes are compared from the same initial state. The diagnostic requires a reference to end at least 0.02 m closer and to have lower actual-encoding goal MSE by more than 10^{-6}. Predicted reversal requires the opposite predicted preference beyond the same tolerance. All comparisons, ties, and actual-metric disagreements are retained by reference type. Actual intermediate images at steps 5 and 10 are used only in a final-call oracle intervention. Privileged references and diagnostic futures never reach CEM.

The runtime pins JEPA-WMs, RoboCasa, robosuite, and MuJoCo 3.2.6 (Terver et al., 2026; Nasiriany et al., 2024; Zhu et al., 2025; Todorov et al., 2012); commit identifiers are in the package. Recorded physics state is restored, followed by controller-reference cache refresh. Historical replay still differs at millimeter scale. Inference uses FP32, mathematical attention, TF32 disabled, and fixed 16-candidate padded microbatches. The encoder component and official retrained predictor are strictly loaded. The source run permits at most 48 hours on one RTX 4090; completion triggers a CPU audit, not another GPU experiment. Adopting the full published search parameters does not reproduce the entire published evaluation protocol. The task is reaching near an object, not grasping or full kitchen manipulation.

Table 10: RoboCasa outcomes on the same ten valid-goal sources. Native and random use three seeds; zero and geometry run once. Original-goal outcomes use the first seed only and must be compared with the matching native first-seed subset. Lower distance is better.

Controller	Terminal success	Anytime success	Mean distance (m)
Native, valid goal	14/30	24/30	.211979
Random	4/30	11/30	.486181
Zero	0/10	0/10	.272880
Privileged geometry	10/10	10/10	.146839
Native, original goal (first seed)	3/10	5/10	.226634

The frozen primary criterion uses a positive lower paired-bootstrap bound. Its +33.33 percentage-point success difference has interval [3.33,63.33] and passes that criterion. The exact test gives $p = .109375$; neither result is suppressed or substituted for the other. The stronger final-distance difference is -.274202 m [-.338877,-.211517], $p = .001953$, and remains secondary. The valid-minus-original first-seed success difference is zero [-.30,.30], $p = 1$; its distance difference is -.000321 m [-.029086,.033672], $p = .994141$. These outcomes do not establish a benefit from changing goal frames.

Table 11: All valid-goal first-plan diagnostic comparisons, with each denominator retained. The actual-cost check is applied only after the 0.02 m physical-advantage condition. Reversals with actual-prefix input retain the same qualifying pairs.

Reference	All	Physically better	Actual-cost disagreement	Reversal	With actual prefix
Zero	30	7	3	4/4	2/4
Random	30	6	3	3/3	0/3
Privileged geometry	30	29	16	13/13	6/13

All 16 geometry-reference metric disagreements are strict sign disagreements, not numerical ties. The 13 qualifying comparisons come from five source episodes, and the six remaining oracle reversals from three. Repeated optimizer seeds are not additional environments. The zero-variance empirical reversal fraction does not establish a population probability of one. The same extraction retains all 33 original-goal reference comparisons and their small qualifying denominators in machine-readable form.

Post hoc, ten of the sixteen valid-goal terminal failures had previously met the physical criterion, including at a five-step observation boundary; six never met it. This classifies observed trajectories only. It does not evaluate an image-based success detector, a deployable termination rule, or the counterfactual outcome of holding the actuator still. The 15-step diagnostic and 60-step closed-loop outcome answer different questions and are not pooled.

F.4 Reproduction package and coverage

The accompanying package has three distinct levels. **Saved-data analysis** recomputes completed statistics and decisions from packaged arrays using NumPy/SciPy, preserving failures. **Figure reproduction** rebuilds the vector mechanism diagram and quantitative plots from their saved inputs and scripts; layout changes do not change scientific denominators. **Full experiment reruns** additionally require external checkpoints, licensed datasets, compatible environments, and GPU/simulator execution. Retained source snapshots and configurations support this last level but are not evidence that an outside clean-machine rerun has already completed.

The package manifest maps evidence to files and records content hashes. A CPU entry point verifies the manifest and launches available analyses; a separate figure command reproduces plots. Source groups and episode identifiers are retained to reproduce clustering. Anonymization removes machine-specific paths and credentials, not failed outcomes. External weights, gated videos, and simulator assets are not redistributed. Their origin, required revisions, and acquisition prerequisites are documented. Completed RoboCasa aggregate records, pairwise diagnostic margins, trajectory-distance extraction, and the independent CPU arithmetic check are included. Raw HDF5 inference and simulation are not rerun by that check. The completed follow-ups are included in the current package and mapping; full raw simulation and inference require external assets.

F.5 Independent confirmation: full selection and decision rules

The physical endpoint score is

$$e = \max\left\{ \frac{\sqrt{\|p - p_g\|_2^2 + \|o - o_g\|_2^2}}{20\,\mathrm{px}}, \frac{|\,\mathrm{wrap}(\theta - \theta_g)|}{20^\circ} \right\}.$$

Here p is the pusher position and (o, θ) the object's position and orientation. Positions use the simulator's 512-by-512 coordinate system, including when localized from resized 224-by-224 RGBs. Fine success requires $e < 0.25$; coarse success requires $e < 1$. The positional threshold applies to the combined norm, not independently to both positions. We also report pusher position error, object position error, wrapped angle error, and object-only fine success. These endpoints are not standard PushT coverage success or anytime success.

The coarse criterion follows the public DINO-WM PushT endpoint predicate: a combined position norm below 20 pixels and a shortest angular error below 20 degrees. Executing the pinned public predicate on all 80 saved development outcomes gives exactly the same coarse classifications as our score. Fine success tightens both scales by a factor of four. This relation concerns the endpoint predicate only. Public CEM invokes a simulator evaluator during search and may stop when its entire batch succeeds; its MPC configuration permits further rounds. Its replay-derived goal also retains terminal velocity, whereas our RGB-goal interface specifies zero velocity. Our search and execution budgets are fixed, with separate source selection and observation interfaces. These scores therefore do not reproduce the complete published benchmark. Public endpoint predicate, planning configuration, public CEM

The completed functional pilot contains 16 development sources and one optimizer seed per source. The analytic baseline and component analysis were added after inspecting that pilot; they retain their exploratory status.

The independent confirmation protocol was fixed before its model scores were inspected. It excludes all 139 source groups used in the active diagnostic, training, and feedback-development chain, uniformly chooses 24 from the remaining 46, and uniformly selects an eligible time index within each chosen group. Eligibility requires sufficient preceding commands, a legal 25-command continuation, and visible current and goal pusher inputs. No model score is used for selection and failed sources are not replaced. Before planning, 22 of the selected 24 sources require object translation of at least 5 pixels or wrapped rotation of at least 5 degrees.

The confirmation ran two optimizer seeds per source and all four neural arms. The analytic controller was executed once per source. Its outcome is reused in paired comparisons, never counted as two independent executions. Statistics first average optimizer seeds within each source, then bootstrap the 24 source units with 10,000 draws. The frozen rule requires feedback to reduce mean joint error by at least 10% against every comparator, with a negative paired upper confidence bound; attain a fine success rate of at least 25%; increase fine success by at least 12.5 percentage points with a positive lower bound; and avoid a decrease in the coarse-success point estimate. Each optimizer seed must separately improve mean error and fine success counts against every comparator.

The predefined object-changing subset is reported separately. A claim about improved joint control on tasks requiring object movement additionally requires positive lower bounds for the fine-success gain against native and analytic control within this subset. These tests do not establish improved object dynamics. All results and failed criteria are reported below; the completed experiment contains 216 unique executions and 5,400 commands.

F.6 Earlier independent precision and execution baselines

F.6.1 Independent precision comparison

On 38 opened DINO-WM witness comparisons, feedback ranks 20 correctly, versus one for native, two for endpoint correction, and 11 for spatial consistency; witness visual MSE falls by 26.95%. We then evaluate newly searched plans on 24 sources excluded from the active development chain, with two seeds per neural arm.

Table 12: Independent experiment: 24 source groups, two optimizer seeds per neural arm. Counts are executions; inference uses the 24 sources. Analytic control runs once per source. The coarse-success safeguard fails against endpoint correction and spatial consistency.

Method	Mean joint error	Fine	Coarse
Native	0.779747	4/48	40/48
Endpoint correction	0.650079	9/48	42/48
Spatial consistency	0.762315	1/48	43/48
Known-state feedback	0.507964	23/48	41/48
Analytic pusher control	1.987407	3/24	8/24

Feedback-minus-endpoint mean error is -0.142115 [-0.220267,-0.058473]; fine success improves by 29.17 [14.58,45.83] percentage points. The complete rule fails because coarse success is below endpoint and spatial controls. Its object-position difference against endpoint, -0.8795 [-1.8487,+0.2060] pixels, is uncertain. A second round retains the precision benefit and supports object-position improvement, but still fails the spatial-control coarse safeguard. Full results, the RGB-calibrated cohort, and failed criteria remain in the appendix. These controls reject a purely terminal-score explanation for every effect, without identifying more accurate contact dynamics.

F.6.2 Stronger execution baseline

For the known linear actuator $s_{t+1} = As_t + bu_t$, a T-command finishing sequence solves a minimum-energy endpoint problem. With $\mathcal{C}_T = [A^{T-1}b, \ldots, b]$, full row rank gives

$$u^* = \mathcal{C}_T^\top (\mathcal{C}_T \mathcal{C}_T^\top)^{-1} (s_g - A^T s_0). \tag{19}$$

Admissibility and achieved precision are checked empirically; the derivation does not guarantee a bounded solution. Execute a 25-command first plan followed by this common 25-command rule. It controls final self-state tolerance, not identical finishing commands or contact history.

On 50 public tasks from 20 episodes, native planning followed by analytic finishing attains 29/50 fine successes, versus 22/50 for two-round feedback. Mean joint error is .374859 versus .533268, paired difference -.158409 [-.302807,-.055183]; the +14 [-2.27,31.25] percentage-point fine difference is uncertain. This exploratory strong baseline changes the comparison: latent feedback is not the only source of precision gains.

F.7 ESTIMATING THE ACTUATOR GAINS FROM RECORDED RGB

We separately test whether the two PD gains must be supplied exactly. This calibration study uses only previously opened development sources and did not alter the control confirmation. Eight sources were assigned to calibration and eight to validation before fitting. Calibration uses one native plan per source, its initial RGB, four later RGB observations, and recorded commands: 32 future RGB labels in total. A single bounded least-squares fit estimates the two gains, retaining the PD family, simulation timestep, action scale, and prefix initialization as known structure. Simulator states and validation targets do not enter the fit.

The fitted gains are 102.1442 and 20.4151. On 40 existing plans from the eight other sources, predicted position has RMSE 0.6784 against RGB localization, with a source-bootstrap 95% interval of [0.6094, 0.7515]. Velocity RMSE against simulator velocity, loaded after the fit is frozen, is 0.3698 [0.3420, 0.3978]. All six prespecified calibration criteria pass. Position is measured in the simulator's 512-by-512 coordinate system, whereas the images contain 224-by-224 pixels. The original actuator gains yield position RMSE 0.6918 against those localized RGB targets; the small difference does not establish improvement over the true dynamics because both predictions start from a noisy pixel estimate. A supplementary check against simulator position gives RMSE 0.4640 for calibrated gains and 0.4864 for exact gains, with overlapping intervals.

This result supports estimating two parameters within the known actuator family from a small record of motion. The following control experiment uses these frozen gains consistently for current velocity estimation and future propagation, without retaining exact gains in the initial observation interface.

F.8 CONTROL WITH FROZEN RGB-CALIBRATED GAINS

After the 24-source studies, we evaluated all 22 remaining source groups outside the active 163-source development and evaluation chain. Twenty-one require object motion. Each neural arm uses two optimizer seeds and two rounds of 25 commands; analytic control runs once per source. Command 50 is the fixed primary endpoint, command 25 is secondary. Source resampling averages both seeds first. The cohort has no paired exact-gain arm and is not a pretraining holdout.

At command 50, feedback, native, endpoint correction, spatial consistency, and analytic control have mean joint errors 0.486243, 0.763940, 0.621565, 0.780820, and 2.108498. Fine counts are 17/44, 2/44, 8/44, 1/44, and 3/22; coarse counts are 41/44, 36/44, 40/44, 39/44, and 8/22. Feedback's fine-success gains against native and endpoint correction are 34.09 [15.91, 52.27] and 20.45 [2.27, 38.64] percentage points. All intervals are pointwise.

The full rule fails because paired mean-error intervals against every neural comparator include zero. The differences against native and endpoint correction are -0.277697 [-0.557330, 0.053291] and -0.135322 [-0.476000, 0.224384]. Coarse point estimates and both seed-direction checks pass; these do not override the failure. Pusher-position differences support improvement, but object-position and angle differences against neural controls include zero. At the secondary 25-command endpoint, feedback achieves 18/44 fine successes; its complete rule also fails.

A post hoc source inspection retains all outcomes: feedback improves mean error over endpoint correction in 17/22 sources and worsens it in five. One seed at source 169 has an 83.02-pixel object-position error and 107.16-degree angle error. This is a realized control failure, not an excluded

outlier or an identified consequence of calibration. The sources and parameters change between cohorts, so their average differences cannot isolate the effect of replacing exact gains.

F.9 Detailed second-round extension and stopping reconstruction

We subsequently extend every trajectory of the independent 24-source experiment by one further 25-command plan, retaining its first round unchanged. Each controller receives its own actual RGB after command 25 and its complete executed-command history, with the original goal, cost, and actuator interface. All neural arms receive another 300-candidate, 30-iteration search with paired random streams; analytic control replans once per source. The endpoint is fixed at command 50, without oracle stopping or best-time selection. This is an extension on already opened sources, not a new source-independent confirmation. Its thresholds, methods, seeds, and budget were fixed before the new outcomes were opened.

Table 13: Two rounds on the same 24 sources. All methods receive a second observation and another 25-command execution; neural search budgets remain equal. These are opened-source extension results. The overall rule still fails against spatial consistency's coarse success count.

Method	Mean joint error	Fine	Coarse
Native	0.750306	6/48	40/48
Endpoint correction	0.576453	7/48	44/48
Spatial consistency	0.607157	1/48	45/48
Known-state feedback	0.374006	23/48	44/48
Analytic pusher control	1.972640	3/24	8/24

The precision advantage persists against every comparator and in both optimizer seeds. Against native and endpoint correction, feedback reduces mean joint error by 50.15% and 35.12%, with paired differences -0.376300 [-0.578253, -0.221067] and -0.202447 [-0.268429, -0.135911]. Fine-success gains are 35.42 [12.50, 56.25] and 33.33 [12.50, 54.17] percentage points. Feedback and endpoint correction now have identical coarse classifications in all 48 paired executions. The full rule still fails, however: feedback has 44 coarse successes against spatial consistency's 45. Two losses occur in one source group, partly offset by one gain in another. Neither the original failure nor this remaining failure is removed.

More execution does not monotonically preserve success. Feedback gains five fine successes and loses five, leaving 23/48 unchanged. Its mean error nevertheless decreases by -0.133958 [-0.333274, -0.011437] from command 25 to 50. A secondary difference-in-changes analysis finds an increased error advantage over native of -0.104518 [-0.201396, -0.012731]; the corresponding intervals against endpoint correction and spatial consistency include zero. These changes combine additional observation, search, and execution, so they do not isolate the causal benefit of reobservation. The appendix plots both endpoints for all methods.

Executed object-position error also falls relative to endpoint correction by -1.5874 [-2.5041, -0.7533] pixels. Its angle-error contrast, -0.6986 [-1.6192, 0.1267] degrees, includes zero. These pointwise secondary comparisons extend the endpoint evidence, not a claim that unknown contact dynamics are modeled more accurately. This experiment adds 5,400 physical commands and retains the same 24 statistical source units.

A post hoc protocol check applies the public evaluator's coarse-success stopping rule to the same saved trajectories, with a two-round cap: retain command 25 if coarse success is reached there, otherwise use command 50. It never selects the smaller error. Fine successes are 23/48 for feedback, 4/48 for native, 9/48 for endpoint correction, 1/48 for spatial consistency, and 3/24 for analytic control. Feedback's mean-error differences against native and endpoint correction are -0.2987 [-0.4805, -0.1710] and -0.1687 [-0.2331, -0.1081]. Its coarse count remains 44/48 against spatial consistency's 45/48. Thus the precision gain persists in this particular stopping analysis, but the coarse-success limitation remains. This is an opened-source reconstruction using physical-state success signals, not a new independent experiment, a deployable visual stopping detector, or a reproduction of unlimited public MPC. Both original failed rules remain unchanged.

A further experiment freezes RGB-calibrated actuator gains and evaluates all 22 remaining sources outside the active 163-source chain. Feedback attains 17/44 fine successes, native 2/44, and endpoint correction 8/44. Fine-success difference intervals are positive, but mean-error intervals include zero and the complete rule fails. The appendix reports this separate cohort; it is not pooled with the 24-source experiments.

F.10 New temporal-window validation of the corrected-first composition

The selection and decision rule are frozen before model outcomes: exclude every earlier seed-99 and 16-episode window by at least 26 start indices, retain all ten eligible episodes, and choose one start uniformly per episode. Seven goals require object motion; no initial state succeeds even coarsely. This validates new temporal conditions within the same episode bank, not new environments. The common analytic finish follows each arm's own first plan and state; its commands need not be equal across arms.

The primary endpoint is object-only error, the maximum of object-position error divided by 20 pixels and wrapped angle error divided by 20 degrees. Seeds are averaged within episode before 10,000 paired bootstrap draws. A positive claim requires an upper 95% error-difference bound below zero in each model. Every final pusher-position error must be at most one pixel and velocity error at most one pixel/second. Additional bounded checks require nondecreasing fine/coarse point rates, nonincreasing joint P90, and lower mean object error in both seeds. These tolerances are not trajectory equality or population equivalence.

LeWM completes all checks: object error falls from .505946 to .264364, paired difference -.241582 [-.514021,-.027857]; joint error falls from .507088 to .264481. Fine success changes from 10/20 to 11/20, with an uncertain +5 [-15,30] percentage-point difference; coarse success changes from 17/20 to 20/20 and joint P90 from 1.604217 to .514491. All arms' maximum pusher-position and velocity errors remain below the specified tolerances. Twenty coarse successes do not establish zero population failure risk.

DINO-WM completes but fails the object-specific comparison: object error changes from .117665 to .175916, paired difference +.058251 [-.011038,.155128]. Fine success falls from 18/20 to 17/20; coarse success remains 20/20, and joint P90 rises from .232127 to .281477. Mean object error increases in both seeds. Final self-state tolerances pass. The interval does not establish harm, but it does not support the proposed improvement, and the point safeguards also fail. Thus the two-model primary claim is not supported.

A post hoc episode sensitivity check gives LeWM seven improving, two worsening, and one tied episode; DINO-WM has two improving, seven worsening, and one tied. Removing any one episode preserves the sign of each model's mean difference: negative for LeWM, positive for DINO-WM. This is a descriptive check, not another significance test or permission to alter the primary decision. The interventions differ across models, so the discrepancy does not isolate an architecture effect. Both full phase executions and 54 independent statistics per model pass verification; execution success does not erase the scientific failure.

F.11 Fixed-budget observation-frequency development test

The LeWM component result motivates one bounded test on the same 16 opened episodes and two seeds. We retain 50 executed commands and 60 CEM updates, comparing two rounds of 25 commands with 30 updates per round to ten rounds of five commands with six updates each. The planning horizon remains 25 commands. Both the native objective and terminal constraint are tested. This redistributes search and observations together, increases encoding overhead, and does not isolate observation frequency or match exact FLOPs.

The terminal-constraint variant deteriorates: mean joint error rises from 1.402991 to 2.418928, fine success falls from 6/32 to 0/32, and coarse success falls from 18/32 to 2/32. The paired joint-error increase is +1.015937 [.378565, 1.659876]. Object-position error has a negative point difference of -6.9620 pixels, but its interval [-14.8953, 1.8634] includes zero; pusher error increases by +33.4702 [26.2679, 40.9157] pixels. The native short-interval arm also has 0/32 fine and 2/32 coarse successes. The specified conditions for spending on DINO-WM transfer fail, so this schedule is not expanded or tuned.

A saved-plan inspection, requiring no further model or simulator calls, finds that the terminal-constrained controller's five-command execution matches its known-actuator position projection within .0490 pixels on average. Its projected goal-position error averages 73.3198 pixels at command five and 11.8957 at command 25; the unexecuted suffix contains much of the anticipated progress. The original matched initial searches also retain substantially higher cost after six updates than after 29. These observations motivate under-optimization and a repeatedly deferred terminal objective as possibilities, but do not separately identify their effects. More frequent real observations alone cannot be credited with solving reliability under this fixed allocation.

F.12 V-JEPA2: CORRECTED BASELINE AND THE LIMITS OF ACTION-PERMUTATION TESTS

The JEPA-WM design study already identifies a two-step rollout-loss implementation problem in the original released V-JEPA2-AC and supplies a retrained baseline (Terver et al., 2026). We therefore compare the original and official retrained predictors using the same saved encoder features, actions, targets, and 48 windows from 24 DROID test episodes (Khazatsky et al., 2024). This is a development robustness check on already opened windows. The retrained model also removes proprioceptive input, so the difference does not identify the effect of fixing one bug. The cache lacks source timestamps and uses a center-cropped image interface; this is not a reproduction of the official 4-Hz planning protocol.

The fixed primary comparison uses two-frame context and step four. Original and retrained autoregressive L1 are .408495 and .414555, with paired retrained-minus-original difference +.006060 [-.000612, .012261]. Neither establishes a benefit over copying the initial feature. We retain both context windows and both history conditions below; observed history is an evaluation intervention requiring recorded intermediate images.

Table 14: Common-reference V-JEPA2 development comparison. Copying uses the initial feature for predicted history and the latest observed feature for observed history. These are offline losses, not control outcomes.

Predictor	Context	History	Factual L1	Factual minus copy
Original	2	Predicted	.408495	+.010945
Original	2	Observed	.357225	+.008304
Original	4	Predicted	.399864	+.002314
Original	4	Observed	.341522	-.007399
Retrained	2	Predicted	.414555	+.017005
Retrained	2	Observed	.369712	+.020791
Retrained	4	Predicted	.401950	+.004400
Retrained	4	Observed	.355830	+.006909

Copying uses the initial feature for predicted history and the most recently observed feature for observed history. All predicted-history copy comparisons have intervals containing zero. Factual inputs often fit the observed target better than zero or cross-episode actions, but the alternative actions' actual outcomes are unavailable. Such scores measure factual conditional matching, not counterfactual correctness or optimal action choice.

A post hoc action audit explains a further limitation. Reversing these action sequences changes the intermediate path but preserves the summed Cartesian displacement. For all 48 windows, the integrated final XYZ difference is at most 2.98e-8 meters; the gripper difference has the same maximum after the official per-step clipping. The final orientation difference averages .1224 degrees, with 47/48 below one degree. The maximum within-path Cartesian difference averages 1.2324 centimeters. Thus the intervention does not ensure a meaningfully different terminal pose. Near-identical commanded poses do not prove identical physical object outcomes, but without those outcomes, failure to prefer the factual sequence at the terminal image cannot certify an action-order deficit.

Observed-history comparisons raise a distinct issue: they retain the factual sequence's intermediate images while substituting alternative action inputs. They therefore differ from executing that alternative sequence from the original state. A preference change between predicted and observed histories alone does not identify accumulated latent error as its cause. These boundaries motivate using actu-

ally executed, outcome-distinct action comparisons for functional claims, as in the feasible-witness diagnostic. This offline comparison supplies no evidence of V-JEPA2 robot-control improvement.

F.13 VERIFICATION RECORDS AND ANALYSIS SCOPE

The retained experiment artifacts include frozen configurations, source selections, commands, saved observations and states, predictions, optimizer records, and process exit records. Each of the independent experiment and its second-round extension was checked by replaying 216 physical trajectories, 192 public final model calls, and 16 complete optimizer runs; 91 source-level statistics per endpoint were separately recomputed. The extension also verifies every new starting input and preservation of all first-round results. The calibrated cohort adds 198 trajectories, 352 public model calls, 32 complete optimizer replays, and 182 independently recomputed statistics. A self-contained NumPy/SciPy analysis package reproduces the reported diagnostic, development, calibration, confirmation, extension, and calibrated-control numbers, including failed criteria, secondary trajectory changes, and the post hoc stopping reconstruction. It does not reproduce checkpoint inference or simulation. The local working records have not been publicly released.

The public-task and subsequent expansion artifacts are separately verified: 126 local statistics for public control, 77 for its composition baseline, 98 per model for the 16-episode comparisons, 252 for the corrected-first composition development, and 348 for the corrected V-JEPA2 comparison. These additions and the subsequent controlled comparisons are included in the current candidate package; the earlier archive is retained only as a historical version. Appendix F.32 gives the current mapping. The subsequent ten-episode validation is complete, with 54 independently verified statistics per model and its failed two-model claim retained.

F.14 KNOWN ACTUATOR TRANSITION

This subsection refers to the earlier RGB-localized protocol. Its original exact-gain actuator uses $k_p = 100, k_d = 20$; position initialization is subject to RGB localization error. The later calibration and common-finish studies use the fitted gains listed in Table 9. Appendix A.7 is the common specification of the transition matrix, zero-based indices, command displacement, and endpoint solution. The subsystem propagates pusher self-state; it contains no object contact model or future observation.

F.15 LEWM TERMINAL CONSTRAINT IN THE PUBLIC-TASK EXTENSION

LeWM has no independently writable proprioceptive channel in the evaluated checkpoint. Its native visual terminal cost is augmented with squared known-state errors: pusher-position squared error divided by 20 squared, plus velocity squared error divided by 20 squared. The multiplier is the median native cost divided by the median state cost on the initial candidate batch, with a 1e-12 denominator floor; it is frozen for that planning call. This scaling uses no executed outcome or future image. Raw action proposals and execution are clipped to ±1, retaining LeWM's interface. All later composition experiments reuse this rule unchanged. DINO-WM retains its distinct native action convention and latent-feedback intervention, so the two systems do not form a one-factor architecture comparison.

F.16 FROZEN-ROLLOUT IMPLEMENTATION

This subsection specifies the earlier RGB-localized protocol, including its zero goal velocity and clipped action proposals. The generated common-finish experiments retain the same state-overwrite operation but use recorded current/goal proprioception and their public action convention (Table 9).

Let N_s be the checkpoint's state normalization, E_s its existing proprioceptive encoder, and B the operation that broadcasts one state embedding over every visual patch. For a candidate of 25 raw commands, actuator propagation gives states s_{5k}, with embeddings $q_k = E_s(N_s(s_{5k}))$, for $k = 0, \ldots, 5$. The initial latent uses the actual RGB and q_0. Only one actual image initializes a planning call; earlier executed commands determine velocity, not extra visual history.

At chunk k, the predictor receives the available latent history, truncated to its configured history length. Let its next visual and patchwise proprioceptive outputs be $\widetilde{v}_{k+1}$ and $\widetilde{q}_{k+1,i}$. The appended observation channels are

$$z^{\text{obs}}_{k+1,i} = (\widetilde{v}_{k+1,i},\ q_{k+1})$$

for known-state feedback. The spatial consistency control instead broadcasts $\widetilde{q}_{k+1,0}$ at intermediate boundaries. Native and endpoint correction retain the original patchwise outputs there. Before another prediction, the action channels are replaced by the next candidate action chunk. At the fifth boundary, all three corrections use q_5 for the terminal state term, while native uses its original predicted state.

With goal embeddings (v_g, q_g), every arm minimizes the same sum of two mean squared errors,

$$J(U) = \frac{\|\widehat{v}_5 - v_g\|_F^2}{D_v} + \frac{\|\widehat{q}_5 - q_g\|_2^2}{D_s},$$

where D_v and D_s count scalar entries in their respective embeddings. The goal state is encoded from its RGB-localized pusher position and zero target velocity. Physical joint error is an evaluation measure and does not enter this objective.

CEM starts with zero mean and unit standard deviation in normalized action coordinates. Each iteration draws 300 Gaussian candidates and replaces the first candidate by the current mean. Candidates are mapped to raw commands, clipped componentwise to $[-1, 1]$, and normalized again before both neural prediction and known-state propagation. The 30 lowest-cost candidates determine the next normalized mean and sample standard deviation, without momentum or a variance floor. After 30 iterations, the clipped final mean is executed; the method does not choose the best previously sampled trajectory. Every arm uses batches of 50 and the same initialization, bounds, update rule, and iteration count. The planner has no environment object, simulator query, or success-based early stopping.

F.17 Fixed-pair development outcomes

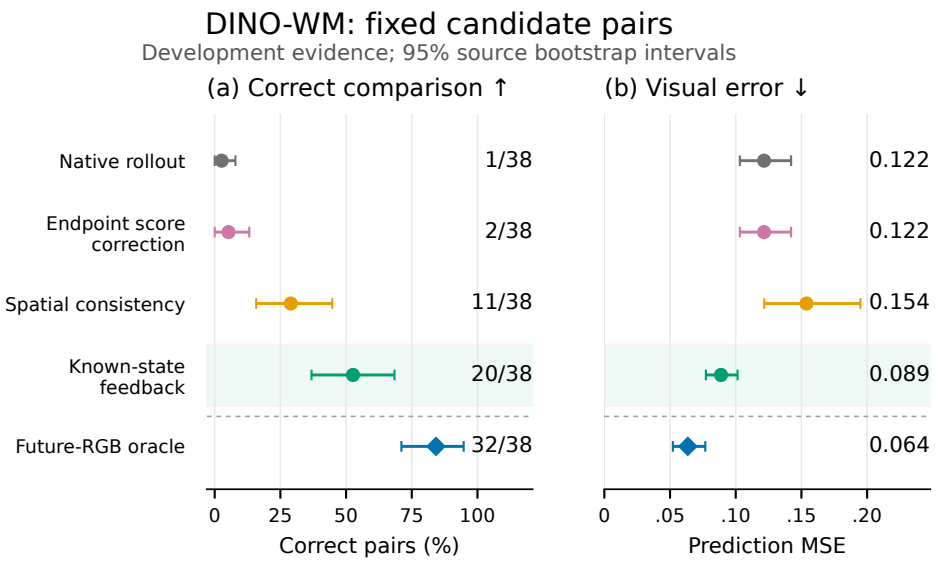


Figure 16: Fixed-pair development on 38 DINO-WM comparisons. Known self-state changes intermediate inputs and improves both ranking and visual prediction error relative to the controls. Actual intermediate RGB is an evaluation oracle. The source-bootstrap intervals describe these opened development comparisons; they are not independent method confirmation.

F.18 Development control outcomes

Table 15: Functional development on the same 16 sources. Fine and coarse successes use the terminal joint criterion in Section 4. All four neural planners have identical coarse success counts. The analytic baseline is an additional exploratory comparison.

Planner	Mean joint error	Fine /16	Coarse /16
Native	0.611982	0	15
Endpoint correction	0.467706	1	15
Spatial consistency	0.598779	1	15
Known-state feedback	0.345530	9	15
Analytic pusher control	1.104221	5	10

Feedback reduces mean joint error by 43.54% relative to native CEM. The paired difference is -0.266452, with a source-bootstrap 95% interval of $[-0.388470, -0.143216]$. Relative to the analytic pusher controller, the mean error decreases by 68.71%; however, the fine-success difference has an interval of [0, 0.50], so the pilot does not establish a strictly positive lower confidence bound for that success comparison.

The gain has a specific scope. Pusher error decreases from 10.384 to 3.540 pixels, whereas mean object position error changes from 4.626 to 4.771 pixels and its paired interval includes zero. Five of the feedback controller's nine joint successes occur among the 12 sources requiring object motion; the remaining four start with the object already within its positional and angular tolerance. The analytic controller places the pusher within tolerance in all 16 cases but achieves only five joint successes. Together, these results support a development hypothesis about joint precision using self-state and visual constraints. They do not show that object dynamics prediction has generally improved.

The public coarse endpoint criterion gives 15/16 successes for every neural arm in this pilot. Their difference becomes visible at the predefined fine criterion. Thus the development gain concerns terminal precision within a setting where most native plans already satisfy the coarser endpoint

requirement. The public-task study tests whether the difference persists after additional planning and execution under a shared budget.

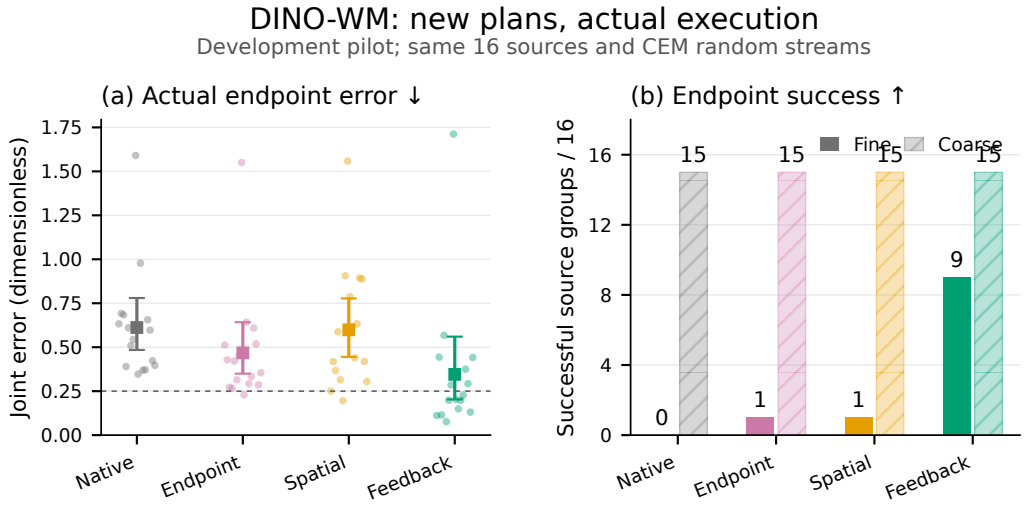


Figure 17: New CEM plans on 16 development sources. The plot retains all source outcomes for four neural planning variants. Known-state feedback improves joint precision, with the component analysis locating the main average gain in pusher position. Table 15 includes the additional analytic baseline. These are development results; Table 12 reports the independent experiment.

F.19 CONDITIONAL FEASIBLE-REFERENCE OUTCOMES

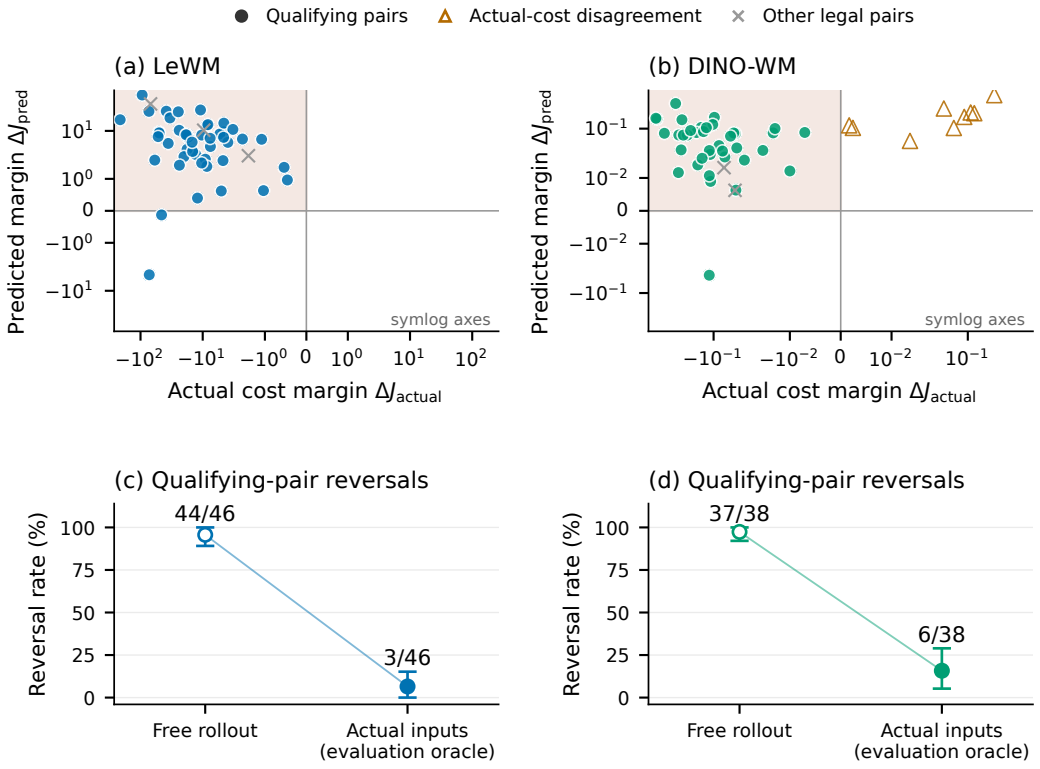


Figure 18: Feasible-witness comparisons in both systems. (a,b) show all 49 legal witnesses; margins are witness cost minus that model's own CEM cost. The shaded region has opposite actual and predicted preferences. Filled circles additionally satisfy the prespecified physical-advantage and actual-reference conditions. Triangles retain the nine DINO-WM comparisons with physical advantage but without the required actual-reference preference; crosses are other legal pairs. Axes are symmetric logarithmic, in each model's native cost units. (c,d) use the unchanged 46- and 38-pair denominators and pointwise source-bootstrap intervals. Actual intermediate encodings reduce reversals but require future observations: they are an evaluation oracle.

F.20 INDEPENDENT SOURCE-LEVEL OUTCOMES

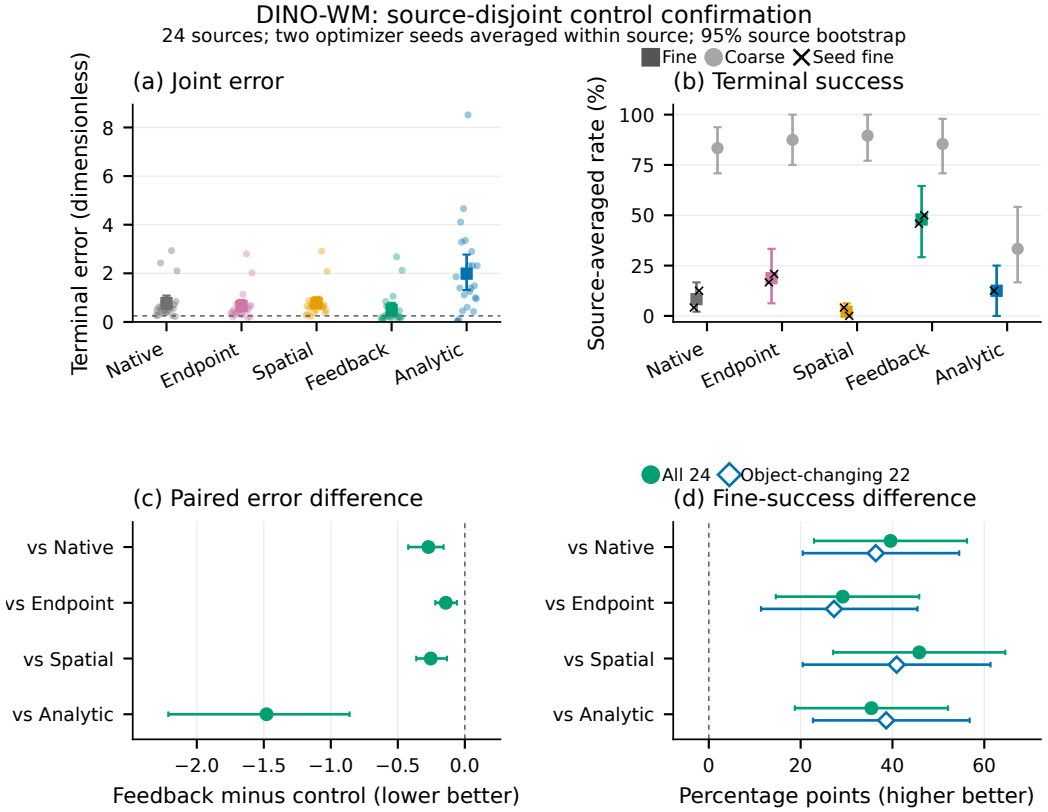


Figure 19: All independent sources and controls. Seeds are averaged within source before bootstrap resampling; crosses in (b) give individual seed fine-success rates. (c,d) show paired feedback differences, with the predefined 22-source object-changing subset also shown in (d). Intervals are pointwise. Precision improves, but the overall rule fails because coarse success is lower than two controls.

F.21 INDEPENDENT COMPONENT OUTCOMES

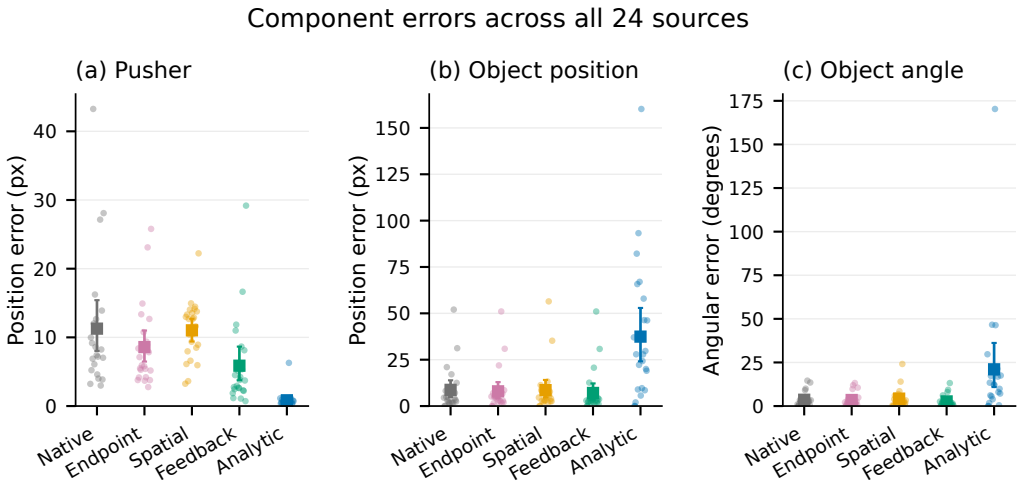


Figure 20: Independent component errors, retaining all 24 source means and all five controls. Position uses simulator coordinates; components were not selected by their outcomes. Error bars are pointwise source-bootstrap intervals. The analytic controller precisely positions the pusher but often misses the object goal. Joint and object-only conclusions remain distinct.

F.22 Outcomes after the second planning round

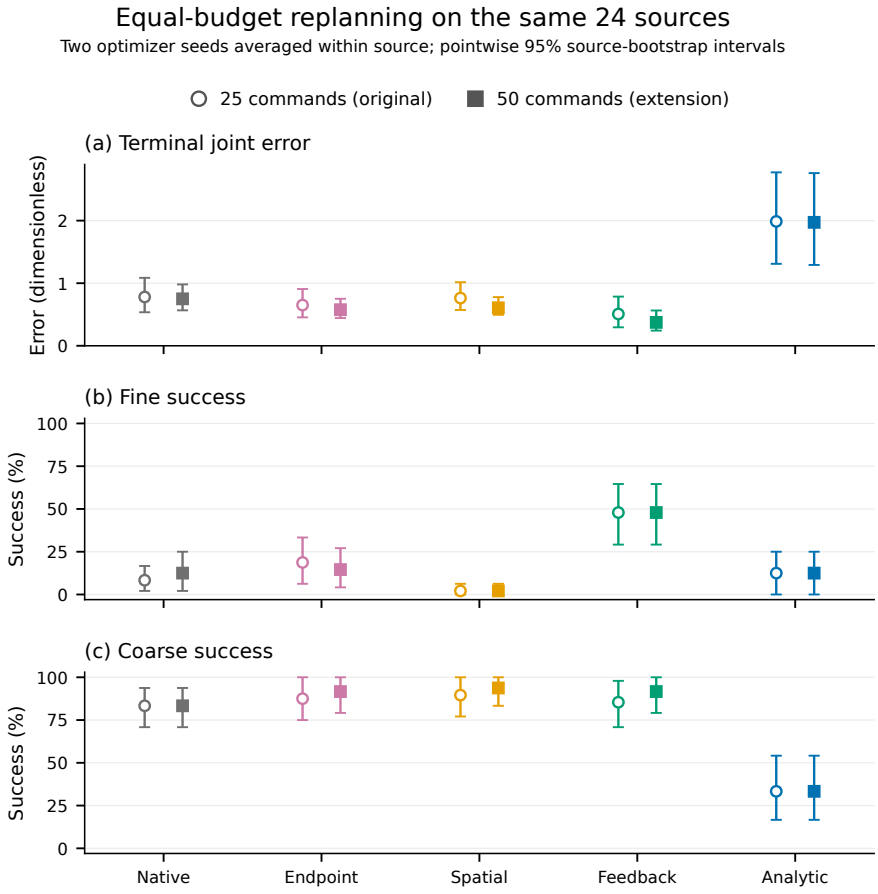


Figure 21: First and second endpoints of the same trajectories, at 25 and 50 commands. All five methods retain the original 24 sources; neural methods use two seeds, and analytic control runs once per source. Intervals resample sources after averaging seeds. Feedback retains its fine-success advantage but gains and loses five individual fine successes. Coarse success equals endpoint correction at command 50 but remains one below spatial consistency. Both complete experimental rules remain failed. Additional observation, search, and execution co-vary; the figure does not isolate the effect of reobservation.

F.23 CALIBRATED-CONTROL ENDPOINT OUTCOMES

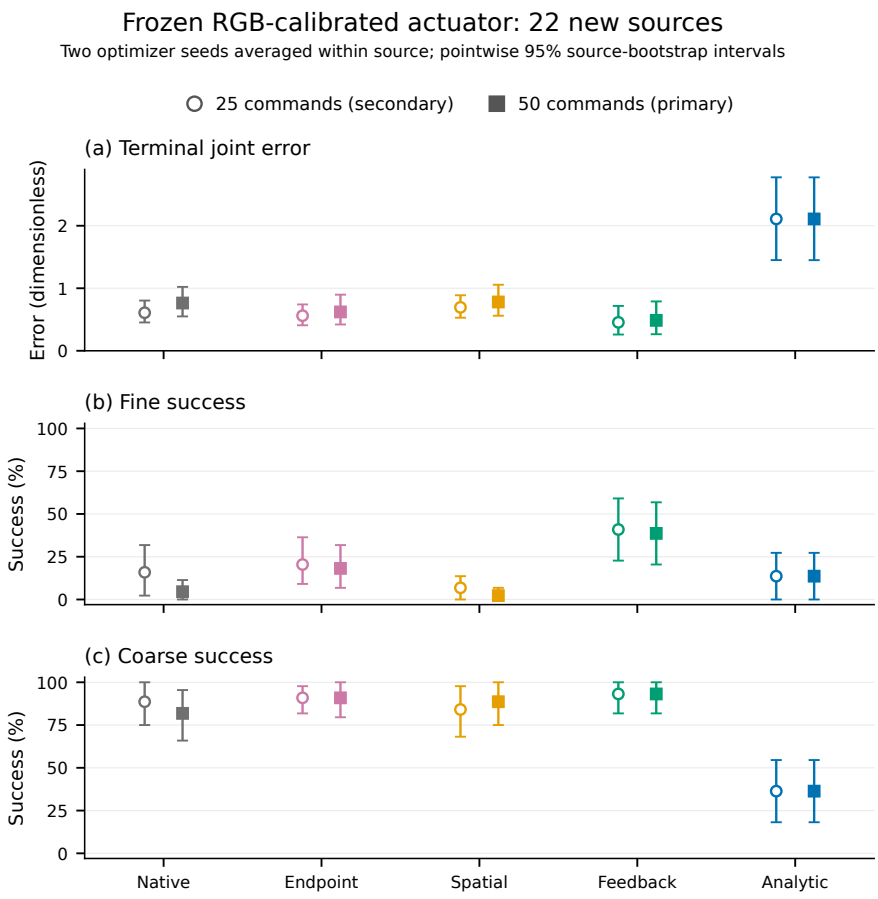


Figure 22: Frozen RGB-calibrated gains on all 22 remaining active-chain source groups. Two optimizer seeds are averaged within source before resampling; analytic control has 22 actual trajectories. Command 50 is primary and command 25 is secondary. Fine success improves, but mean-error difference intervals against the neural controls include zero, so the full primary rule fails. No failed case is removed and no exact-gain arm is paired on this cohort.

F.24 Cross-model composition and remaining failures

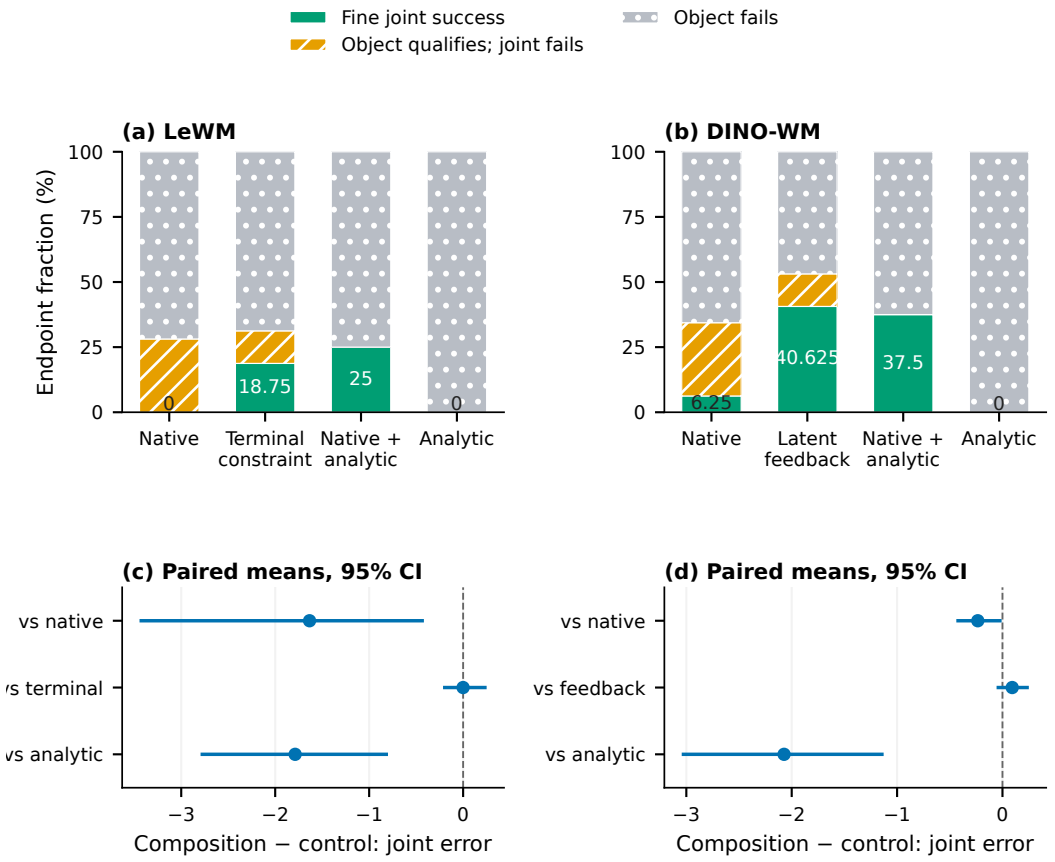


Figure 23: Completed 16-episode, two-seed validation, at command 50. Both models improve over native planning under native-then-analytic composition, while their full comparison rules fail. The stacked bars separate joint fine success, fine object state without joint fine precision, and object-state failure. Analytic control is executed once per episode. Paired intervals average seeds before resampling episodes. This figure precedes the corrected-first development test and does not contain the subsequent ten-window validation.

F.25 EARLIER COMMON-FINISH TEMPORAL WINDOWS

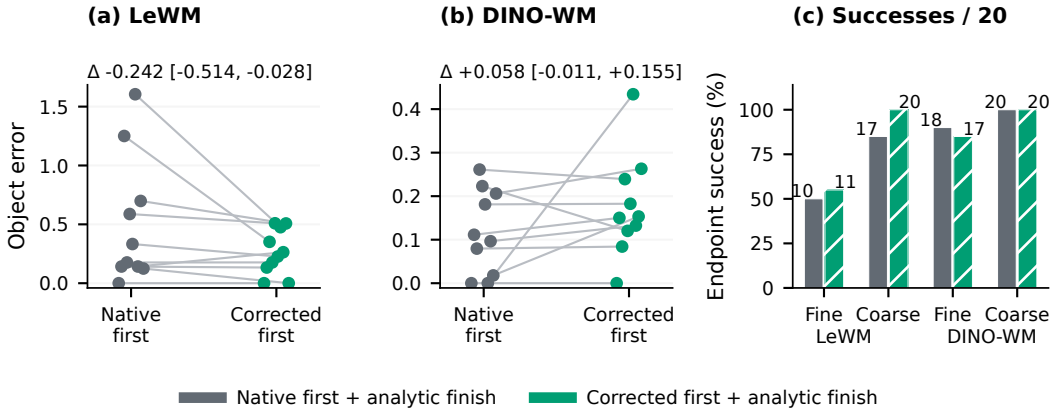


Figure 24: All ten reserved temporal windows and two seeds. Each line in (a,b) connects an episode's seed-averaged object error; axes have different scales. The annotations give corrected-first minus native-first paired mean differences with episode-bootstrap 95% intervals. Both arms use the same analytic finishing policy and satisfy final self-state tolerances. LeWM improves, while DINO-WM's interval includes zero and its point effect worsens. Panel (c) gives executed joint success counts, not population guarantees. The two-model primary claim and several DINO-WM safeguards fail.

F.26 FULL-SEARCH ROBOCASA CONTROL AND COMPARISON PLOT

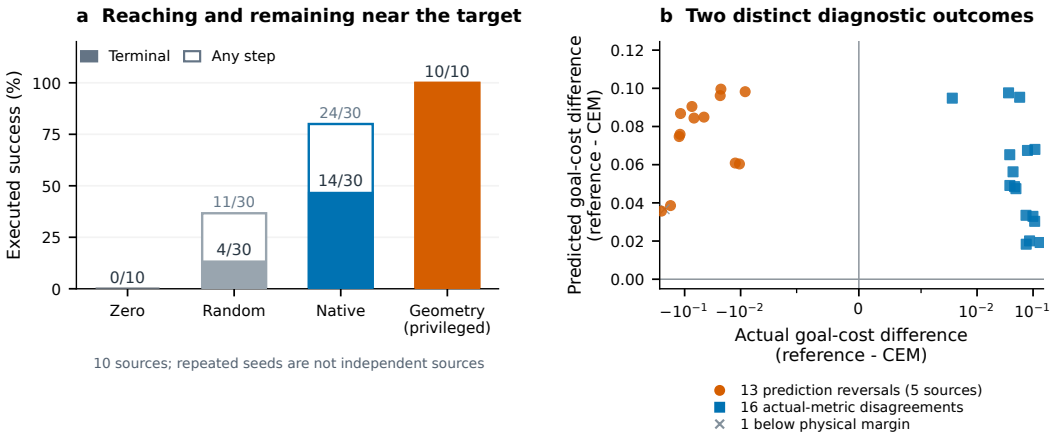


Figure 25: Completed V-JEPA2 reaching and diagnostic outcomes. (a) Filled bars show terminal success; outlines show any-step success, on the same ten sources. Labels are execution counts. (b) Each point compares a fixed first CEM plan with a privileged reference over the same 15 executed steps. All 30 pairs are retained: 13 qualifying prediction reversals, 16 physical improvements opposed by the actual visual cost, and one below the physical margin. Positive differences favor CEM. The horizontal axis is symmetric logarithmic; the vertical axis is linear. Optimizer-selected pairs do not estimate a random-pair error rate.

F.27 RoboCasa trajectory retention

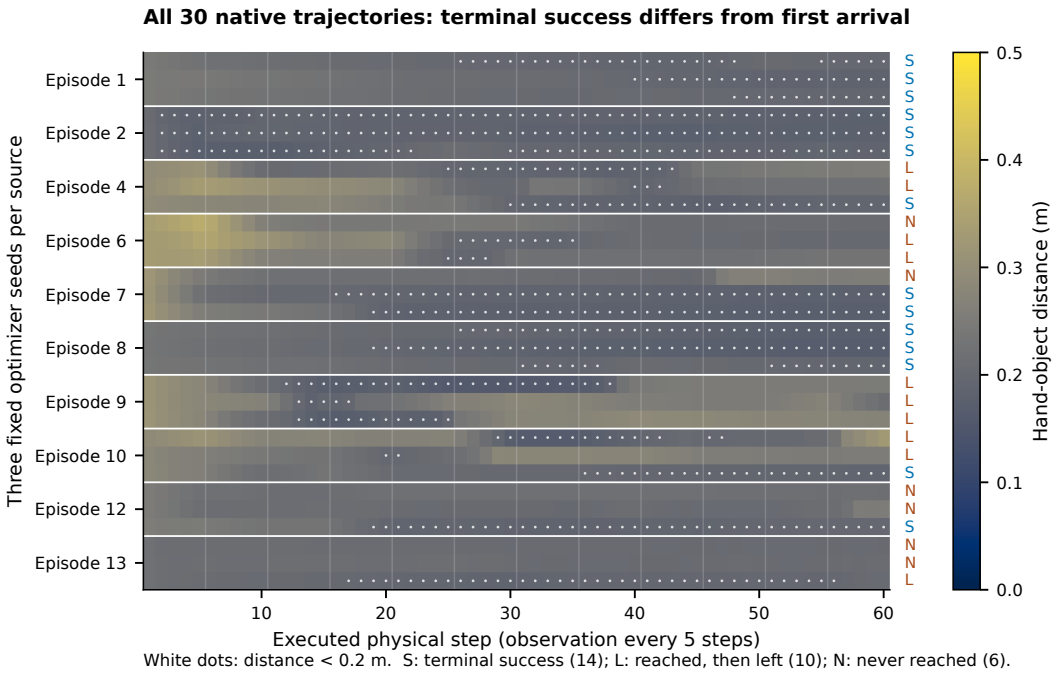


Figure 26: All 30 valid-goal native executions, ordered by source and optimizer seed. Color represents hand–object distance; white dots mark physical success. The right labels distinguish terminal success (S), arrival followed by a terminal failure (L), and no arrival (N). Observation boundaries occur every five steps. Ten leave-after-arrival failures all include a successful observation boundary. This is a descriptive post hoc audit, not an evaluated stopping intervention or a replacement for terminal success.

F.28 RECORDED SIMULATION EXAMPLES

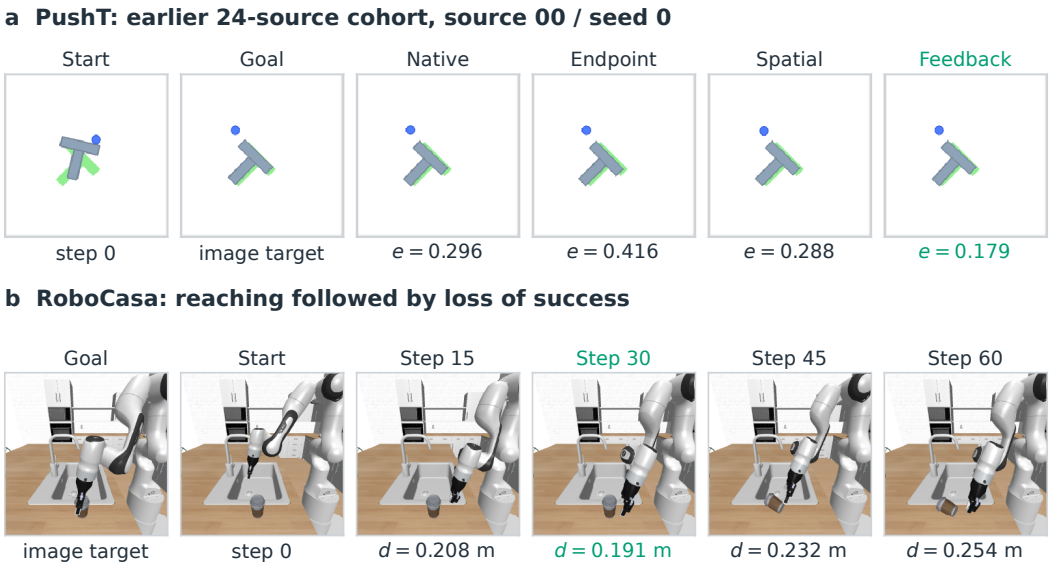


Figure 27: Actual recorded simulator images. (a) The first stored independent PushT source and first seed; all four methods are shown at command 25, with measured joint error. (b) The first source/seed, in numerical order, among RoboCasa trials that reach and then leave the success region (demo 4, seed 1). It satisfies the 0.2 m distance criterion at step 30 but fails at step 60. This post hoc category example illustrates a failure, not a stopping intervention. Full source/seed identities, step-indexed images, all 24 PushT sources, and all 30 RoboCasa endpoints are in Appendix C.

F.29 COMPLETE ROBOCASA BANK DECOMPOSITION

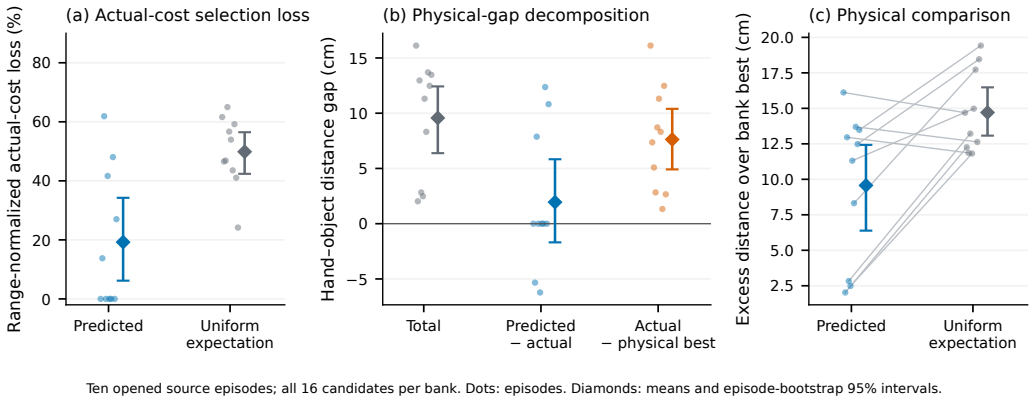


Figure 28: Same-bank decision usefulness and remaining loss, ten opened RoboCasa sources. (a) Range-normalized actual-cost selection loss versus the exact uniform-selection expectation. (b) Step-15 physical gap to the bank's best candidate, decomposed into a signed predicted-versus-actual selection difference and an actual-cost selection gap. (c) Prediction has less excess physical distance than uniform selection. Dots are episodes; diamonds and intervals are episode means and pointwise bootstrap 95% intervals. Physical axes use centimetres. These are finite-set comparisons, not unique causal failure shares.

F.30 COMPLETE GENERATED-TASK FINISHING COMPARISON

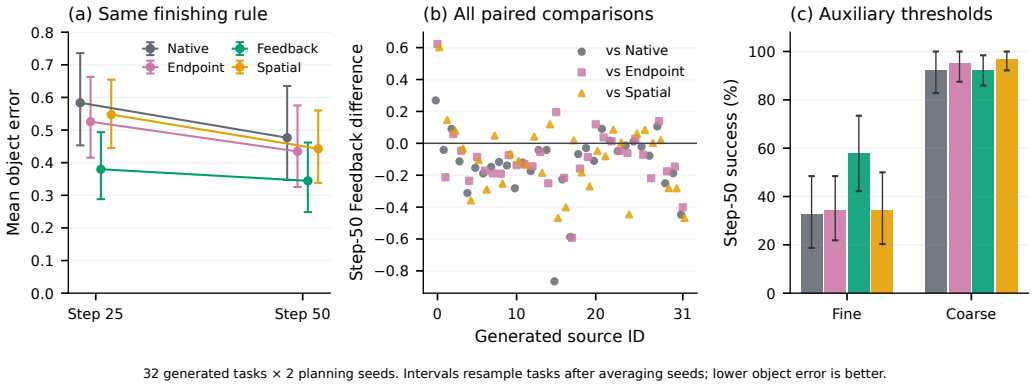


Figure 29: All 32 generated tasks and four arms, with seeds averaged before inference. (a) Object error before and after the same finishing rule. (b) Every task's feedback-minus-native, endpoint, and spatial differences. The added spatial contrast is -.0984 [-.1726,-.0175]. (c) Fine success improves, but feedback has fewer coarse successes than spatial. All 256 executions are retained; the added arm is not a new confirmation cohort.

F.31 ADDITIONAL CONTROLLED COMPARISONS AND THEIR COMPLETE OUTCOMES

F.31.1 COMMON CANDIDATE BANK AND DECISION LOSS

Ten fixed banks are built on the previously opened RoboCasa episodes. Each contains zero action and 15 sequences sampled before actual outcome inspection, with three legal seven-dimensional commands, each repeated for five physical steps. Every source uses its original paired goal and reset. The action bank was fixed before candidate outcomes were inspected. The regret, uniform-reference, and physical-decomposition analyses were specified after the original results were opened. It contains only zero and random proposals, rather than a representative sample from every possible search distribution. All 160 candidates, all 1,200 full-bank pairs, and the overlapping 1,050 random-only pairs are provided.

For each candidate the records contain original predicted cost, actual-encoding cost, actual-prefix-oracle cost, hand–object distance, and error to the recorded goal EEF position. Argmin ties follow the first stored index. Pair comparisons use a 10^{-6} actual-cost tolerance; predicted and oracle differences within that tolerance do not count as reversals. Every source contributes one normalized decision loss, dividing R_J by that bank's actual-cost range; a zero range is assigned zero. Both signed physical differences and actual-cost losses are retained. Bootstrap intervals use 10,000 resamples of ten source episodes, seed 2026092451. Pairs are never resampled as independent environments.

For zero plus random, mean actual-cost loss is .041779 [.011188,.079334], normalized .192380 [.061825,.342504]. The actual-prefix oracle has loss .046697, with paired oracle-minus-original difference +.004918 [-.012511,.020672]. The change in non-tied pair misranking is -.021667 [-.063333,.018333]. These results do not support reduced finite-bank decision loss from the oracle. The signed physical gap for original prediction is +.019498 m [-.016896,.058328]. The random-only subset is an overlapping sensitivity analysis, not replication. Its mean decision loss is .048535. The packaged candidate vectors independently reproduce every selection and all pairs. Original raw bank distance checks occurred before export; the offline package does not repeat simulation or feature inference.

F.31.2 SAME-BANK UNIFORM REFERENCE AND PHYSICAL DECOMPOSITION

This additional analysis was specified after the original A/B/C results were opened. It uses their saved candidates and endpoints, adds no dynamics or model calls, and is labeled post hoc. For A, the physical variable remains hand–object Euclidean distance after the original 15 physical steps. Uniform selection is integrated exactly over the same 16 candidates; it is not another sampled policy

rollout. The random-only result excludes only the predetermined zero candidate and remains an overlapping sensitivity analysis. All intervals average within source and use the original 10,000 episode-bootstrap draws and seed 2026092451; signed paired contrasts additionally report the exact two-sided sign-flip test over ten sources.

Table 16: Full-bank decision losses and physical gaps. All ten source episodes contribute; the physical minimum is a hindsight candidate, not a feasible-information policy. Intervals are pointwise.

Quantity (units as labeled)	Mean	95% interval
Native cost units		
Predicted-choice actual-cost loss	.041779	[.011188,.079334]
Uniform expected actual-cost loss	.104305	[.075221,.136540]
Predicted minus uniform actual cost	-.062526	[-.099222,-.027779]
Fractions of each bank's actual-cost range		
Range-normalized predicted loss	.192380	[.061825,.342504]
Range-normalized uniform loss	.498614	[.423778,.564698]
Paired normalized difference	-.306234	[-.460488,-.140248]
Physical distance (metres)		
Total physical gap to bank best (m)	.095709	[.063837,.124258]
Signed predicted-minus-actual difference (m)	.019498	[-.016896,.058327]
Actual-cost selection gap to physical best (m)	.076211	[.049144,.103961]
Uniform expected physical gap (m)	.147063	[.130767,.164784]
Predicted minus uniform physical distance (m)	-.051354	[-.078574,-.022461]

Conditional on sign exchangeability under the null, the enumerated paired tests are $p = .0078125$ for predicted-minus-uniform actual cost and $p = .015625$ for physical distance. The signed predicted-minus-actual physical contrast remains uncertain ($p = .375$). The three physical terms sum separately in every source. A negative signed term is possible; the decomposition must not be converted into percentages of uniquely caused failure. The best physical candidate and best actual-cost candidate both use future outcomes.

The normalization formula in the main text matches the original implementation: divide each source's loss by its own positive actual-cost range, then average sources. Exactly zero range maps to zero; no positive range is clipped or removed. The full-bank range is .125212–.453626, with no range at or below the existing 10^{-6} pair tolerance and no exact or 10^{-6} near-tie at the predicted minimum. Exact argmin uses the first stored index. Raw and normalized losses, all source ranges/indices, and both subsets are exported in `A_by_source.csv`; negative signed physical differences are retained.

F.31.3 New generated tasks with common finishing

Thirty-two specifications are drawn and saved before any model scoring. Object center is uniform in $[200, 312]^2$; approach direction and initial angle are independently uniform in $[-\pi, \pi]$. The pusher begins 100 pixels behind the object along the approach direction, with zero velocity. Twenty commands of magnitude .2 along that direction followed by five zero commands generate the recorded goal image and state. The generator seed is 2026092461. Sources are neither replaced nor filtered using outcomes. All 32 require object change and none initially meets the coarse or fine predicate. Each source trajectory is independently replayed exactly before model evaluation.

Native, endpoint-only correction, and feedback each use planning seeds 2026092462 and 2026092463, offset by $10000i$ for source i. The public DINO-WM interface supplies observed current and goal proprioception. Each arm executes one full 300-candidate, 30-update CEM plan for 25 commands followed by 25 analytic commands. The requested analytic commands are clipped at $[-1, 1]$, but the clipping count is zero. Planning proposals retain the public native convention. All 192 trajectories, 9,600 commands, two stage endpoints, RGB observations, per-command states, and collision-contact counts are saved. An additional 9,600 commands exactly replay the trajectories; they are validation work, not independent experimental samples. Execution and local audit records are summarized in Table 20; they add no independent scientific samples.

The primary difference is command-50 object-only error, feedback minus endpoint correction. Bootstrap draws (10,000) and Monte Carlo sign flips (50,000) first average seeds within the 32 generated sources, using seed 2026092464. A 0.1-unit benefit was declared as a meaningful-effect reference; it is not a replacement success gate for older studies. The primary difference is -.090124 [-.154208,-.019017], $p = .014440$. Its interval establishes a positive benefit in sign, but not a minimum effect of .1. The command-25 object difference, fine/coarse precision, tail errors, and contact summaries are secondary.

Table 17: All generated-task outcomes, two seeds per each of 32 source tasks. Every final endpoint uses the same analytic finishing rule. Fine and coarse counts are joint endpoint precision, not coverage.

Stage	Method	Mean object error	Fine /64	Coarse /64
25 commands	Native	.583600	5	48
	Endpoint	.525537	8	54
	Feedback	.379883	20	58
	Spatial	.547521	2	56
50 commands	Native + finish	.476433	21	59
	Endpoint + finish	.434519	22	61
	Feedback + finish	.344395	37	59
	Spatial + finish	.442795	22	62

Feedback-minus-endpoint fine success is +.234375 [.109375,.359375]; coarse success is -.03125 [-.09375,.03125]. The coarse interval does not establish harm, and no noninferiority margin was retrospectively imposed. Command-50 joint-error P90 is .885846/.722922/.665576 for native/endpoint/feedback. All arms meet the final pusher-position and velocity tolerances. Mean object displacement during finishing is 5.290/4.239/1.762 pixels, with mean object rotation 2.093/1.552/.749 degrees. Recorded contact totals are 12,197/12,028/11,480; these count all simulator collision contacts, not exclusively pusher–object contact. The finish changes object outcomes and does not equate contact history. This new generated distribution does not erase the failed DINO-WM temporal-window comparison or establish pretraining OOD generalization.

F.31.4 Complete same-task comparisons and success transitions

The original B already executed Native, Endpoint, and Feedback on the same 32 tasks. Its fixed primary is Feedback minus Endpoint; the following complete comparison display does not replace that primary. The feedback-minus-native values reproduce the original stored secondary comparisons. Endpoint-minus-native and the gain/loss presentation are post hoc. Every difference is computed task by task after averaging its two seeds, so $\Delta(F - N) = \Delta(F - E) + \Delta(E - N)$ holds for each endpoint. The generator samples distinct states directly; tasks are not windows sampled from a shared parent episode. They still share a fixed simulator, actuator family, source-generation distribution, and pretrained model.

Table 18: Paired command-50 comparisons under the same analytic finishing rule. All intervals use the 32 generated tasks. Fine/coarse columns are probability differences, not percentage units.

Difference	Object error	Fine success	Coarse success
Feedback − Endpoint	-.090124	.234375	-.031250
95% interval	[-.154208,-.019017]	[.109375,.359375]	[-.093750,.031250]
Feedback − Native	-.132038	.250000	.000000
95% interval	[-.208854,-.064765]	[.109375,.390625]	[-.062500,.062500]
Endpoint − Native	-.041914	.015625	.031250
95% interval	[-.127087,.017366]	[-.078125,.109375]	[-.031250,.109375]

Object-error Monte Carlo sign-flip values for the three contrasts are .014440, .000340, and .328473, respectively. Fine-success paired intervals and counts are auxiliary and use no familywise correction. Relative mean error reduction versus Endpoint is 20.74%, and versus Native 27.71%; these are ratios of arm means, not averages of per-task ratios. The previously reported ten-window DINO

comparison used Feedback versus Native, whereas B's fixed primary uses Endpoint. The same-task Native contrast now separates this comparator issue, but does not identify which distribution property explains the earlier non-confirmation.

For Feedback versus Endpoint, fine outcomes are 20 both-success, 17 gains, two losses and 25 both-failure. Coarse outcomes are 57 both-success, two gains, four losses and one both-failure. Versus Native, fine outcomes are 19/18/2/25 in that order and coarse outcomes 57/2/2/3. Endpoint versus Native has fine outcomes 16/6/5/37 and coarse outcomes 58/3/1/2. These are 64 paired execution counts; inferential units remain the 32 tasks. Zero net coarse change versus Native includes two gains and two losses and does not establish noninferiority. Figure 34 displays every transition type. All source-level errors and contrasts, both stages, and per-task transition counts are supplied in `B_by_source.csv` and `B_transitions_by_source.csv`.

No execution is removed for failing a finishing tolerance. All 192 are included. Maximum position errors for Native/Endpoint/Feedback are .035355/.033284/.035411 pixels; maximum velocity errors are .051624/.048290/.038151 pixels/s. All are below the pre-existing one-unit bounds. The added same-task spatial comparison below holds broadcasting fixed while changing the intermediate values. It does not equate contact histories or identify contact-prediction accuracy.

F.31.5 SAME-TASK SPATIAL CONTROL AND FIXED-CANDIDATE SCORING

The added Spatial + Finish arm uses all 32 already observed B tasks, the same two planning seeds and source offsets, the same 300-candidate, 30-update CEM, and the identical analytic finishing rule. At each intermediate boundary it broadcasts the original predicted first-patch state, without averaging or replacing it with propagated values; its terminal known-state cost equals the other corrections' cost. All 64 complete plans, 3,200 executed commands, and 3,200 replay commands are retained. The first-patch rule, seeds, thresholds, sources, and model remain fixed; execution checks are listed in Table 20. This added control does not make the observed cohort independently held out.

The sole added primary is feedback-minus-spatial object error at command 50: -.098400 [-.172630,-.017483], Monte Carlo sign-flip $p = .017160$, using the original 32-task inference procedure. Spatial's mean is .442795 versus feedback's .344395, a 22.22% reduction in arm means. Component differences are -1.082961 pixels [-2.371744,.234743] in object position and -1.476047 degrees [-2.848193,.084073] in angle; these auxiliary intervals cross zero despite the composite-error contrast. Fine success is 37/64 versus 22/64, difference .234375 [.109375,.375000]; 20 pairs succeed under both, 17 are gains, two are losses, and 25 fail under both. Coarse success is 59/64 versus 62/64, difference -.046875 [-.109375,0], $p = .250735$, with 59 both-success, zero gains, three losses, and two both-failure. This does not establish noninferiority or equivalence. Spatial's maximum finishing position/velocity errors are .036531 pixels/.039006 pixels/s. All 256 four-arm finishes meet the original bounds with no exclusion; Spatial's mean finishing object displacement/rotation is 4.035 pixels/1.624 degrees and its total collision-contact count is 9,269. Benefits are not restricted to the pusher term, but no unique contact mechanism follows.

For a shared-action diagnostic, the already executed Native/Endpoint/Feedback plans form 64 three-candidate banks; adding Spatial's plan forms overlapping four-candidate banks. Endpoint, Spatial, and Feedback each score every candidate, with no new search or simulation. All 256 candidates are retained, totaling 768 rollout scores; native and upstream calls are interface checks only. All original own-rule plan costs reproduce exactly. Predicted costs roll out from the saved initial history under each fixed candidate. Actual costs encode that candidate's saved command-25 outcome. Both use the same goal interface and normalization; no candidate's future executed image enters its predicted rollout. This input separation is explicit in `experimental_source/spatial_controls/score_bank.py`. The selected candidate's already executed step-50 physical outcome is reported separately; it is not used to compute prediction error. Each bank belongs to one task/seed, and inference averages the two seeds before task resampling. This is a post hoc audit of optimizer-generated proposals, not an outcome-independent random bank or a new confirmation cohort.

Feedback selects its own optimized candidate in 64/64 three-candidate banks and 61/64 four-candidate banks. The selected physical outcomes therefore largely reuse the original Feedback executions; this is not an independent functional confirmation. In the three-candidate bank, feedback-minus-spatial regret is -.038223 [-.073418,-.001624], $p = .041119$, whereas the selected physi-

Table 19: Fixed-candidate scoring. R_J uses actual encoded cost at command 25; the last column uses the selected plan's physical outcome after finishing. Both overlapping banks and all rules are retained.

Candidates	Scoring rule	Mean R_J	Normalized R_J	Object error at 50
N/E/F	Endpoint	.088147	.679985	.432532
	Spatial	.076348	.562624	.387291
	Feedback	.038125	.239348	.344395
N/E/F/S	Endpoint	.093694	.573865	.424940
	Spatial	.114082	.638116	.445159
	Feedback	.049192	.214240	.343423

cal contrast is -.042896 [-.099526,.021293], $p = .177096$. In the four-candidate bank, these differences are -.064891 [-.113502,-.012187], $p = .018820$, and $-.101736$ $[-.174727, -.022151]$, $p = .012680$. Against endpoint, the corresponding three-candidate differences are -.050022 [-.078684,-.021227] and -.088137 [-.144528,-.032637]; four-candidate differences are -.044502 [-.076903,-.008333] and -.081517 [-.144767,-.012585]. All are pointwise diagnostic contrasts. A rule's optimized proposal is preferentially represented: feedback selects its own candidate in 64/64 original banks and 61/64 expanded banks, selecting Spatial in the remaining three; Spatial selects its own in 57/64 expanded banks. Thus lower same-bank regret supports a scoring consequence of the intervention on these proposals, but does not identify how much of the separately optimized control effect it mediates. Adding Spatial also changes the bank's attainable minimum, so raw regrets across bank sizes are not directly comparable.

Exact argmin ties take the first stored candidate in N/E/F/S order. Each task/seed bank's regret is divided by that bank's positive actual-cost range, then the two normalized seed values are averaged within task before task-level resampling; exactly zero range maps to zero without clipping. Neither bank has a zero or at-most-10^{-6} positive range, or an exact/at-most-10^{-6} predicted-minimum tie. The exact uniform-selection expected regrets are .078166 and .094321 in the three- and four-candidate banks. Full raw-scale and normalized intervals, selection identities and source-level records accompany the result. Local CPU checks recompute 2,048 cost components from all saved features across both runs, with maximum discrepancy 8.10×10^{-8}; the lightweight package retains arrays and these raw-feature audit records, not the full feature tensors.

For stage interpretation, write $\Delta_t = E_F(t) - E_C(t)$ and $\Delta_{50} = \Delta_{25} + (\Delta_{50} - \Delta_{25})$. Against Spatial, $\Delta_{25} = -.167638$ [-.248349,-.081392] and the finishing modification is +.069238 [.020283,.131524], $p = .008940$. The benefit is already present before finishing and narrows afterwards. Against Endpoint, the modification is +.055529 [.003008,.125082], and its Monte Carlo sign-flip value is $p = .065299$. The percentile interval and conditional sign-flip test use different inferential constructions; both frozen summaries are retained without treating them as a numerical contradiction. Against Native it is +.071679 [-.005025,.169813], $p = .128897$. This identity is a descriptive difference of changes, not a causal allocation to the first plan and finish, whose paths and contacts differ. All original B comparisons and earlier failed reliability criteria remain unchanged.

Figure 3 illustrates the first stored task and seed, selected without inspecting effect size. Its views are vector renderings of the saved object and pusher states using the upstream PushT geometry and the recorded generated-task goal. The original RGB's built-in target silhouette is not this generated goal, so it is not used as the goal overlay. Packaged states, selected original frames, and source hashes accompany the editable figure script. The `theory_and_panels` CPU entry checks endpoints and annotations against the saved arrays. This illustration adds no simulation or independent functional evidence; its aggregate panel uses all 32 tasks.

F.31.6 Goal-semantic translation control and measurement repair

Each of the same ten RoboCasa episodes uses seven fixed translation multipliers $[0, .25, .5, .75, 1, 1.25, 1.5]$, with rotation and gripper commands zero. The original recorded goal image remains fixed. All 70 candidates execute 15 physical steps, totaling 1,050 steps. Every within-episode pair is retained. The position comparison requires at least .02 m EEF-goal error difference. The matched subset additionally requires pairwise EEF orientation difference at most

Table 20: Generated-task execution and audit records. All planned tasks/seeds are retained. Budgets and runtime are engineering records, not scientific endpoints.

Stage	Retained execution	Checks	Exit / budget
Original B	192 plans; 9,600 executed plus 9,600 replay commands	5,760 CEM updates; one full-plan repeat; local full-trace audit and 99 statistics	0; 8,954/10,800 s
Added Spatial	64 plans; 3,200 executed plus 3,200 replay commands	1,920 CEM updates; one full-plan repeat; input identities and full traces	0; 3,024/7,200 s
Cross-scoring	256 fixed candidates; 768 rule scores; no search/dynamics	Own-rule costs; 2,048 saved feature-cost components across both runs	Both 0

5°, local fingerpad-position difference at most .005 m, and object-position difference at most .01 m. A secondary check requires the position-better candidate to be no worse in goal orientation by more than 2° or in goal fingerpad error by more than .002 m. Opposition means its actual visual cost is larger by more than 10^{-6}.

The initial program completes data acquisition but fails in analysis: six runtime gripper hinge angles cannot be subtracted from the recorded two-value goal observable, and neither is a metric displacement. We preserve that exit-one result and perform no new candidate execution. Using the saved complete simulator state and original XML, 90 forward-kinematics evaluations recover the two fingerpad collision-center positions in the gripper site's local coordinate system. Their six coordinates, in metres, define the repaired distance. Saved joint values are recovered exactly; goal EEF position is exact. Newly derived EEF position differs from the saved runtime observation by at most .2273 mm; the original saved EEF values remain the positional analysis input. Numerical tolerances are unchanged, but this is a post-collection measurement-definition revision and is disclosed as such.

There are 10 opposing pairs among 168 with the physical margin, 10 among 160 matched pairs, and 10 among 154 with the additional goal-pose check. The ten disagreements arise in five source episodes. Averaging the matched-pair fraction within each episode gives .061905 [.019048,.109524] under episode bootstrap. Raw features and geometry are locally checked for all candidates; all 210 pair classifications reproduce. Unmatched robot configuration, rendering sensitivity, the opened sources, and the revised measurement limit interpretation. The earlier 16/29 hand–object comparison uses a different candidate distribution and physical metric and cannot serve as a before/after rate comparison.

A further post hoc seven-candidate selection audit uses the same recorded-goal EEF-position error and the same physical endpoint, without filtering candidates by pair eligibility. In all ten sources, the actual visual-cost minimum and the position-error minimum select index 6 (translation scale 1.5); the largest tested scale, so their physical selection gap is exactly zero on this truncated support. Each actual-cost minimum is unique. Thus the ten matched pair reversals do not change the top-ranked candidate in these seven-candidate banks. This is a descriptive counterexample to treating pair-reversal prevalence as decision loss, not a population guarantee of goal alignment. It does not negate A's physical gap: A uses different candidates and hand–object distance, while this audit uses recorded-goal EEF position. `C_by_source.csv` records every choice.

F.31.7 Scoring-time controls and interrupted-run preservation

These controls address the scored-horizon versus executed-prefix concern raised by Vesessook & Yang (2026), without claiming a reproduction of that study. Both controls retain the original ten episodes, the original first optimizer seed 2026092401, goals, source resets, full 300-by-15 search, and 60 physical steps. The first-prefix variant scores the first predicted command; the running variant scores the average predicted cost over all three commands. Both execute only the first command for five steps before replanning. Their comparison addresses specific scored-versus-executed horizon choices, not all possible planning strategies.

A user-requested pause retained completed trials and the interrupted prefixes. Earlier timeout-like cleanup text is marked as a user interruption, not a new failure. Recovery reused completed plans

and checked every state and RGB during prefix replay; those checks are not new scientific samples. Parameters, seeds, and thresholds remained fixed.

Table 21: Scoring-time recovery records. Saved prefixes are preserved; CPU checks compare complete traces with the same-seed native baseline.

Arm	Reused plans	Replayed steps	Runtime / remaining cap (s)	Exit
Prefix	42	30	13,833 / 25,867	0
Running	50	10	13,018 / 26,509	0

Scientific results. The native reference below is the paired first-seed subset of ten sources, not the full 14/30 native result. Episode bootstrap and conditional exact two-sided sign flips use those ten sources. Terminal success is native 3/10, prefix 1/10, and running 2/10; anytime success is 6/10, 8/10, and 9/10. Mean final distances are .226313, .243064, and .230780 m. Prefix-minus-native success is -.2 [-.5,0], exact $p = .5$; distance is +.016751 m [-.005762,.038789], $p = .195313$. Running-minus-native success is -.1 [-.3,0], $p = 1$; distance is +.004467 m [-.024982,.029965], $p = .779297$. Neither improves the terminal outcome with the available evidence. Seven trajectories in each changed-objective arm reach and then leave the success region. These are fixed opened-source controls, not a search over settings or proof that every scoring-time alternative fails.

F.32 CURRENT REPRODUCTION MAPPING

Table 22 maps the current package's saved inputs to its CPU checks. This package and `MANIFEST.json` supersede earlier partial archives. Run `run_reproduction.py` for numerical checks; `--figures` also rebuilds the plots and diagrams. These commands do not rerun model inference or simulation.

Table 22: Current evidence-to-analysis mapping. All checks operate on packaged saved data. External reruns need checkpoints, simulator assets, and the pinned environments documented in the package.

Result	Saved input and CPU check	Reproduction boundary
Conditional witnesses and independent precision	Core arrays; `core`	Original raw inference/physics audits retained as records
Common candidate decision loss	All candidate vectors/pairs; `reviewer_controls`, `decision_links`	Recomputes choices, uniform expected loss, physical decomposition and source intervals
Generated common-finish effect	Four-arm endpoint/contact arrays; `reviewer_controls`, `decision_links`, `spatial_controls`	32 tasks; added Spatial control on the observed cohort; full raw audit recorded
Same-bank scoring and stage change	Both candidate-cost arrays; `spatial_controls`	Recomputes both overlapping banks; step-25 regret is distinct from step-50 physical outcome
Supplied-state error sensitivity	All 87-condition cost/state arrays; `state_tolerance`	Recomputes selections and task intervals; partial full-feature audit retained as a record; no new plans or sensor validation
Controlled goal semantics	All 70 geometry/cost records; `reviewer_controls`, `decision_links`	Recomputes all pairs and seven-candidate choices; original XML/FK recovery needs external assets
Scoring-time controls	All 60-step distances for both arms and native; `reviewer_controls`	Local raw HDF5 audit retained; packaged traces reproduce endpoints and paired tests
LeWM fixed-bank audit	Full candidate/features/trajectory records; `lewm_generality` (`analysis/reanalyze_lewm_generality.py`)	Recomputes all choices and paired task summaries; zero regrets and nonnegative-gap inference are explicit
LeWM common finish	Full P1 trajectories, task means and CEM records; `lewm_generality`	Recomputes primary contrast from terminal states; does not rerun dynamics or model inference
Finite-history identities and counterexample	`analysis/verify_finite_history.py`; `finite_history`	Numerical identity checks, not a substitute for proof or measured coefficient bounds
Earlier common-finish scope	Per-model validation arrays; `expansion`	Same-episode temporal windows, all prior failures retained
RoboCasa original full search	Completed comparisons and distance traces; `robocasa_complete`	New simulation is not reproduced by a CPU statistics check

Threshold curves are descriptive, retaining the original thresholds and primary endpoints. Intervals are pointwise except for the labeled within-error-family sensitivity bands in Appendix G.

For a threshold τ, the plotted empirical rate is

$$\frac{1}{N}\sum_{i=1}^{N}\left[\frac{1}{R}\sum_{r=1}^{R}\mathbf{1}\{e_{ir}<\tau\}\right].$$

Indicators are formed for individual seed executions before averaging seeds and tasks; the plot does not threshold a seed-averaged error. This is the implementation in `gen_fig_final_controls.py`.

F.33 GOAL-SEMANTIC CANDIDATES

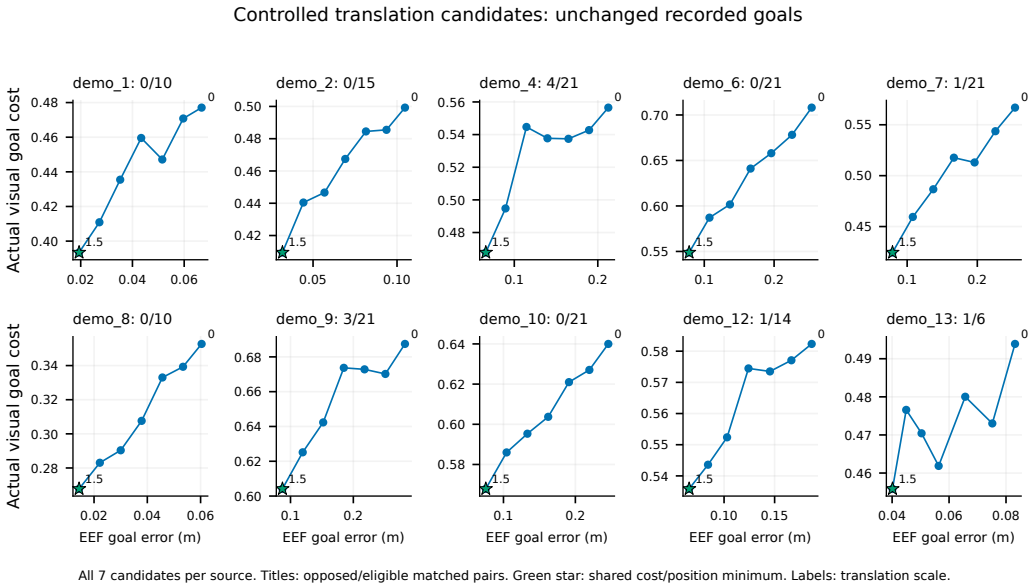


Figure 30: Every fixed translation candidate in all ten opened sources, under the original recorded goal image. Lower EEF-goal position error and actual visual cost are preferable; a local reversal in their direction need not indicate missing physical information. Titles give opposing/eligible matched-pair counts; point labels identify zero and 1.5 translation scale. All seven candidates per source are plotted, including candidates outside particular matched pairs. Green stars mark the shared minimum of actual visual cost and EEF-position error: the shared minimum occurs at the largest tested scale, 1.5, in every source despite local pair reversals. The fingerpad measurement was repaired after acquisition; matching does not cover the full robot configuration.

F.34 ENDPOINT DISTRIBUTIONS AND SCORING-TIME CONTROLS

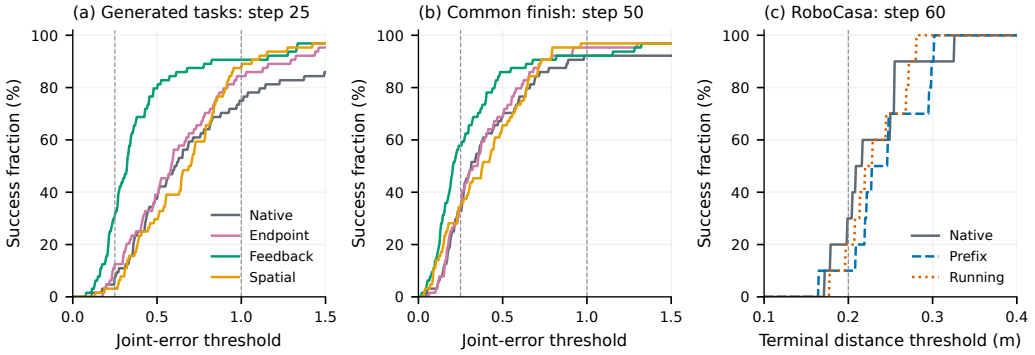


Figure 31: Descriptive endpoint threshold curves, without new threshold selection. PushT indicators are computed per seed execution, then averaged within source and across sources; vertical guides retain the original fine/coarse criteria. RoboCasa curves retain all ten paired first-seed outcomes for the original terminal, prefix, and running objectives, with the original 0.2 m guide. The original primary endpoints and failed rules remain unchanged; no simultaneous significance claim is inferred from curve separation.

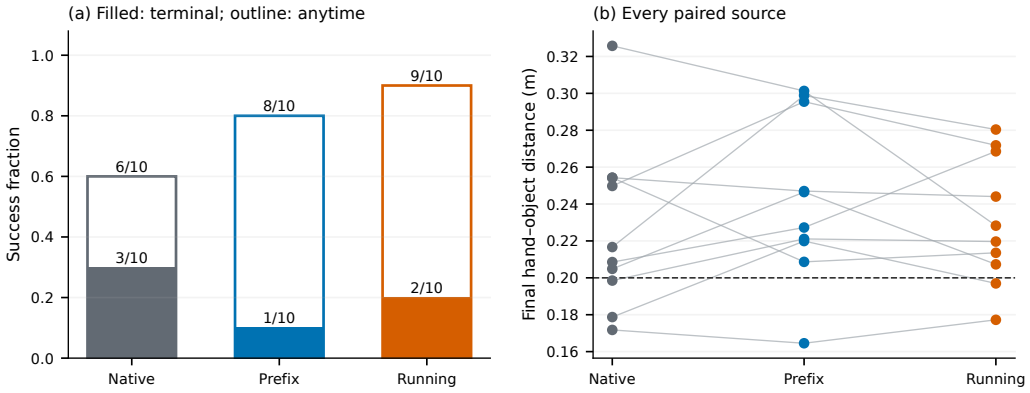


Figure 32: Completed scoring-time controls on the same ten sources and first optimizer seed. Left: terminal and anytime successes. Right: every paired terminal distance, linked to the same native baseline. Prefix and average-horizon scoring do not establish better terminal control; arriving at the threshold is different from remaining there. Both experiments preserve their interrupted prefixes and complete the original fixed search budgets.

F.35 SCHEMATICS FOR THE ADDITIONAL CONTROLLED COMPARISONS

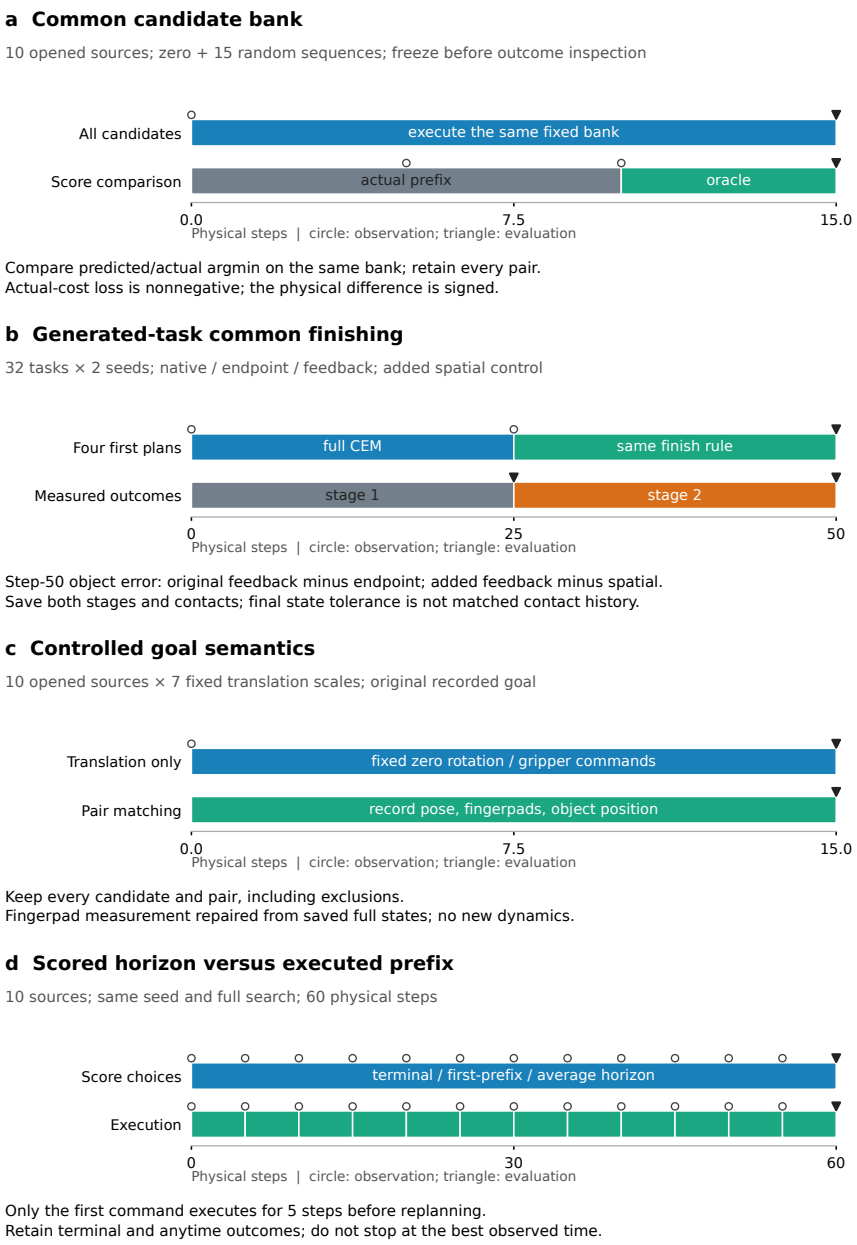
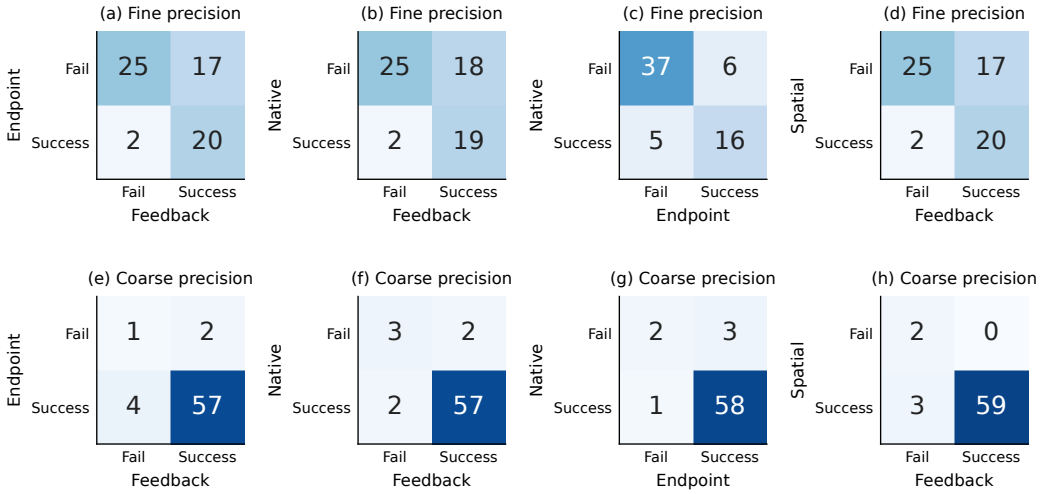


Figure 33: Principles of the additional comparisons: a common action bank identifies finite-set decision loss; common finishing separates a terminal precision explanation from residual object benefit; controlled translation checks specified goal-semantic covariates; scoring-time controls change which predicted horizon is scored while retaining the same executed prefix. Open circles mark observation boundaries and filled triangles mark evaluation times; the primary outcomes are stated in the text.

F.36 SUCCESS GAINS AND LOSSES AFTER COMMON FINISHING



Step 50, common finishing; all 64 paired executions per comparison. Statistical inference uses 32 tasks.

Figure 34: All paired command-50 success transitions on the 32 generated tasks and two seeds, with no tolerance-based exclusions. Rows indicate the comparator and columns the changed method; the upper-right cell gains a success and lower-left loses one. These 64 execution pairs per comparison are descriptive counts, not independent environments; intervals use 32 task means. Equal net coarse counts can conceal both gains and losses.

G SENSITIVITY TO SUPPLIED-STATE ERROR

G.1 FIXED-BANK DESIGN AND INFORMATION INTERFACE

This development sensitivity study reuses all four executed proposals in each of the 32-task, two-seed DINO-WM banks from Appendix F.31.5. The 256 candidates and their actual outcomes are unchanged. The 87 error conditions were fixed before their new scores were opened. No CEM search, simulation, training, or calibration fitting is performed. This is not an independent cohort or a noisy-controller deployment test.

Perturbations are applied to the propagated position or velocity before the original frozen state encoder at each of the five prediction boundaries. Initial observed history and goal inputs stay unchanged. The same perturbation realization is shared across candidates within a task/seed and across strengths. Predictions use only that initial history, the candidate actions, and the specified perturbed supply. Saved command-25 outcomes enter only the subsequent actual-cost evaluation; saved command-50 states provide the selected candidates' post-finish physical endpoints.

Prespecified perturbations. Position strengths are 1, 2, 5, 10, and 20 pixels in the simulator's 512-coordinate workspace, not pixels of the 224-pixel rendered image. Velocity strengths are 5, 10, 20, 40, and 80 pixels/s. Each variable has four persistent biases (the positive and negative coordinate directions) and four boundary-jitter realizations (two Gaussian five-boundary sequences and their antithetic negatives). Bias magnitudes and each jitter sequence's two-dimensional RMS equal the stated strength. A separate structured perturbation multiplies both calibrated gains by 0.8, 0.9, 0.95, 1.05, 1.1, or 1.2. It preserves their ratio and does not refit them. Including nominal supply, there are $1 + 80 + 6 = 87$ conditions and $256 \times 87 = 22{,}272$ candidate-condition scores. Nominal supply means zero *additional* error, not exact true state: its mean position/velocity RMS discrepancy is 0.100 pixels/0.372 pixels/s.

Matched terminal scoring. Endpoint and Spatial retain their original visual rollouts but receive exactly the same perturbed terminal-state cost as Feedback. Thus their cost differences on each fixed sequence remain in the visual term. An auxiliary analysis instead holds the nominal terminal term fixed for all three rules, isolating intermediate-history perturbations from terminal-score changes. The complete grids and both scoring variants are retained; the auxiliary variant does not replace the primary comparison.

G.2 ENDPOINTS, AGGREGATION, AND UNCERTAINTY

The primary diagnostic is Feedback-minus-matched-Spatial raw actual-cost regret at command 25; Endpoint is a comparator. We separately evaluate the chosen candidate's command-50 object error, fine/coarse success, nominal-choice retention, and degradation relative to nominal Feedback. The object endpoint is the same maximum-normalized position/angle metric as in the main text. Rescoring chooses among saved trajectories; it does not create a new physical execution.

For each error family and strength, we average four perturbation realizations and both planning seeds within each task before 10,000 nonparametric bootstrap resamples (Efron, 1979) of the 32 tasks (seed 2026092602). Each gain multiplier has one realization. Candidates, noise draws, and optimizer seeds are not independent evaluation units. Pointwise percentile intervals accompany approximate simultaneous bands formed from the 95th percentile of the maximum standardized bootstrap deviation across all strengths, including nominal supply, *within one error family and one metric/comparison*. These bands do not jointly cover every family, endpoint, or comparator. No interpolation, monotonicity, equivalence, or universal safety threshold is inferred. The nominal bank contrast differs from the four-controller contrast because rescoring can select another arm's proposal.

Table 23: Selected command-50 object error under supplied-state perturbations. Differences are Feedback minus matched-noise Spatial; lower is better. Intervals use task means. Retention is agreement with nominal Feedback's candidate choice. The complete grid, including all gain multipliers and Endpoint comparisons, is in the reproducible numerical records.

Added error	Δe_{obj}	Pointwise 95% CI	Within-family 95% band	Retained
None	-.1017	[-.1758,-.0251]	—	100%
Position bias, 5 px	-.0791	[-.1460,-.0145]	[-.1558,-.0025]	60.2%
Position jitter, 5 px	-.0863	[-.1594,-.0095]	[-.1698,-.0029]	77.3%
Position bias, 10 px	-.0433	[-.1121,+.0265]	[-.1234,+.0369]	43.0%
Position jitter, 10 px	-.0580	[-.1234,+.0094]	[-.1325,+.0165]	54.7%
Velocity bias, 80 px/s	-.0794	[-.1511,-.0092]	[-.1563,-.0026]	68.4%
Velocity jitter, 80 px/s	-.0982	[-.1721,-.0293]	[-.1761,-.0203]	75.4%

Position error. At the tested 1-, 2-, and 5-pixel strengths, object-error contrasts against Spatial retain negative within-family bands. At 5 pixels, raw regret differences are -.06173 [-.10180,-.02047] for bias and -.06008 [-.10557,-.01045] for jitter; corresponding bands are [-.10926,-.01420] and [-.11238,-.00777]. The object intervals cross zero at 10 and 20 pixels (Table 23; Figure 35). This neither makes 5 pixels a maximum tolerance nor establishes failure at 10 pixels.

The Endpoint comparison is less conclusive. At 5-pixel jitter, the object contrast is -.06567 [-.12600,+.00006], with band [-.13577,+.00442]. A 2-pixel bias comparison also has a band crossing zero although the 5-pixel one is negative; significance cannot define a monotone boundary. Relative benefit is not absence of degradation: nominal Feedback's selected object error is .34342, compared with .36632 for 5-pixel bias and .35944 for 5-pixel jitter. Jitter's degradation is +.01602 [+.00519,+.02794]. Its coarse-success contrast against Spatial is -5.078 percentage points [-10.938,-.391], with a within-family band crossing zero. Coarse-success non-inferiority remains unestablished.

Velocity and gain error. Object contrasts against Spatial retain negative bands at all tested velocity strengths through 80 pixels/s, the tested upper limit rather than an estimated tolerance limit. Joint gain multipliers retain 98.44–100% of nominal choices, but the largest downward/upward changes induce only .699/.463 pixels of mean additional position RMS and 4.760/2.485 pixels/s of velocity RMS. These structured perturbations do not establish robustness to arbitrary 20% dynamics errors. Position and velocity are not matched in encoded norm; their different outcomes do not identify a universal direction-specific law.

Nominal-terminal auxiliary comparison. With the terminal term held nominal, 5-pixel bias yields a Feedback-minus-Spatial object difference of -.08370 with band [-.16348,-.00392]. Jitter yields -.08273 but its band [-.16619,+.00072] crosses zero. The full matched-terminal and isolated-history results are reported together rather than selecting the more favorable variant.

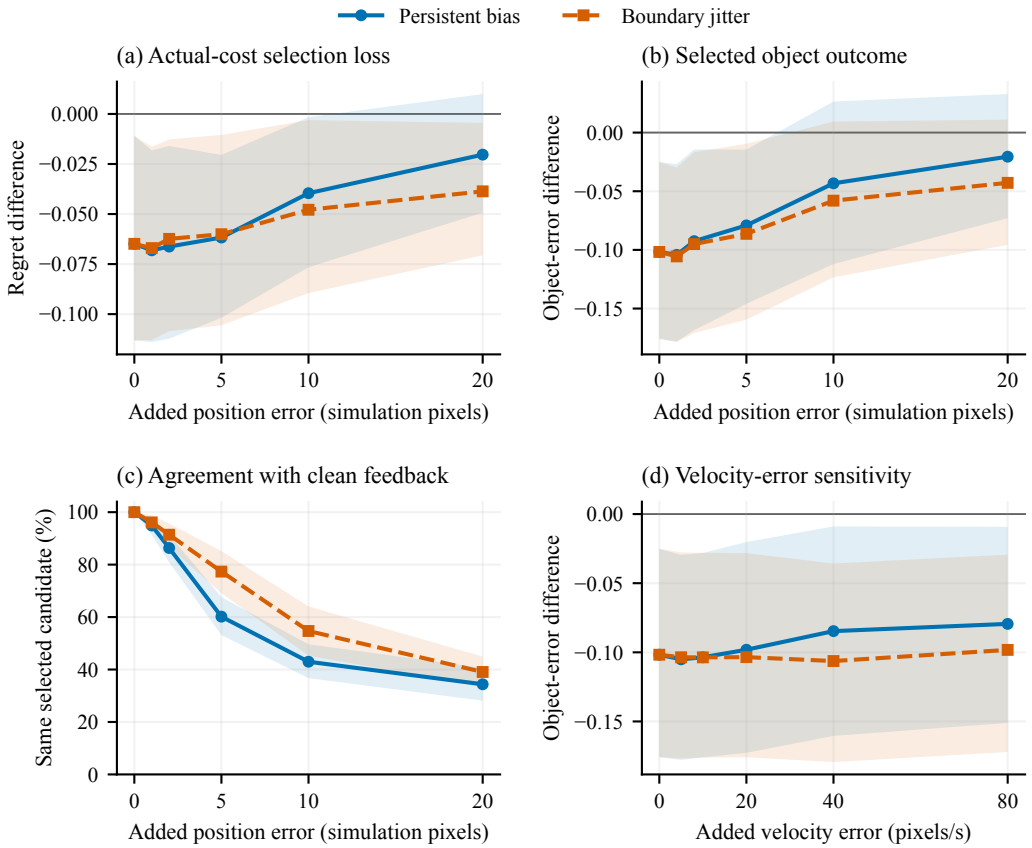


Figure 35: Sensitivity on the observed four-candidate banks. Panels (a), (b), and (d) compare Feedback with Spatial using the same perturbed terminal-state term; negative values favor Feedback. Panel (c) shows retention of nominal Feedback choices. Lines join the prespecified strengths for display, without assuming intermediate behavior; shaded bands are *pointwise* task-bootstrap 95% intervals, not the within-family bands in Table 23. "Clean" denotes nominal calibrated supply with zero additional perturbation, not error-free state. Physical outcomes are reused from the saved executions; there is no new noisy-controller rollout.

G.3 INTERPRETATION AND REPRODUCTION BOUNDARY

Equation 8 retains $(I - A)^{-1}B\boldsymbol{\xi}$: supplied-state error can still propagate through visual history. Its coefficients are unmeasured, so it does not convert the physical perturbation strengths into guaranteed performance limits. Finite-bank decisions additionally depend on cost margins. For nominal choice U_0, define $m_U = \widehat{J}_0(U) - \widehat{J}_0(U_0)$ and $d_U = \widehat{J}_\xi(U) - \widehat{J}_0(U)$. If $m_U + d_U - d_{U_0} > 0$ for every $U \neq U_0$, the choice is unchanged. Common cost shifts cancel. Checking this finite-bank identity on saved costs is not certification for unseen actions.

The application suggested by these observations is an approximate module that propagates self-state from available initial information and candidate actions. Accuracy should be checked by state component and temporal error structure, alongside candidate-level consequences. Real initialization/goal sensing errors, latency, candidate-dependent bias, contact coupling, full CEM replanning, and hardware deployment remain untested. The banks are optimizer-conditioned and already observed; nominal Feedback chooses its own proposal in 61/64 banks. The sensitivity analysis therefore adds a bounded future-application motivation, not independent functional replication.

The package's `state_tolerance` entry recomputes all 27 family/strength summaries, choices, task-level intervals and bands from complete compact cost/supply arrays and saved boundary states. The historical audit checked all 22,272 supply trajectories, 1,792 explicit actuator integrations, nominal-cost parity, and all 1,392 cost components from complete final features on the prespecified task 0 only. Full feature tensors for that partial audit are not in the lightweight package; its audit record is retained. Other tasks do not receive a claimed full-feature audit. GPU execution exited zero within its fixed budget. Packaged CPU reanalysis invokes neither models nor simulation and cannot replace an end-to-end rerun.