Prediction, Action Ranking, and Closed-Loop Control: A Replay-Verified Audit of Visual World Models in PushT

Yicheng Pan

5 October 2026

Abstract

Prediction error, action ranking, and closed-loop success measure different parts of a world-model controller. We examine their relationship in PushT using fixed action banks, controlled planner comparisons, and complete episode replay. On fifty paired scenes, changing only the CEM objective from predicted coverage to terminal pose cost raises success from 11/50 to 33/50. The prespecified exact two-sided McNemar test gives $p = 1.04904 \times 10^{-5}$; the paired mean peak-coverage gain is 0.18187, with a descriptive 95% bootstrap interval of $[0.14816, 0.21329]$. An unchanged native Diffusion Policy succeeds on 25/50 scenes. Contact reranking succeeds on 33/50 versus 25/50 for first-candidate execution, but its mean peak-coverage difference is only 0.00099, with interval $[-0.03346, 0.03851]$. Development experiments explain why separate endpoints are needed: low fixed-bank regret accompanies little closed-loop gain, a richer physics prior changes native choices on only two of 64 contexts, and one optimizer-selected plan predicted to attain 0.75682 coverage attains zero. All 250 confirmation episodes and 59,630 executed actions pass numerical reconstruction checks. We provide the saved decisions, trajectories, and statistical recomputation tools. The findings establish a planner-objective effect within this benchmark; renderer-specific physical priors and unknown upstream policy exposure limit comparisons with learned models and claims of generalization.

1 Introduction

A world-model controller turns observations into forecasts, uses those forecasts to select actions, and executes the actions in an evolving environment. Prediction accuracy measures the first link. Ranking an action bank measures the second. Task success depends on the whole loop, including the planner's objective and the states that its decisions visit. In contact-rich pushing, a small pose error can change predicted overlap with a goal, while a useful intermediate motion may temporarily reduce that overlap.

We study these links in PushT, where a circular pusher must align a T-shaped object with a fixed target. The central experiment holds a contact-based forecaster, CEM search parameters, and the per-decision rollout budget fixed, and changes the planning objective. On fifty locally reserved scenes, terminal-pose CEM succeeds on 33 scenes and coverage-objective CEM on 11. This paired comparison was specified before the scenes were inspected. It isolates an objective effect in the implemented controller without changing its dynamics model.

Two further comparisons put that result in context. First, development intervention banks separate pose error from action-selection regret: more accurate dynamics do not always change the selected action, and accurate ranking within a fixed bank does not establish successful closed-loop control. Second, a five-arm confirmation includes the released Diffusion Policy with its native execution queue and a matched pair of proposal-selection controllers. These references distinguish an objective intervention from changing the policy interface. All five arms retain fifty scenes, including failures.

The study contributes a controlled local comparison and a record of how its predictions and decisions were produced. We save and reconstruct every confirmation action and observation, check paired initial conditions, and recompute the statistical endpoints from raw trajectories. The development experiments remain exploratory because their contexts were reused. The confirmation is prospective with respect to local development, but the upstream policy's training exposure to these seeds is unknown. Physical and learned comparators also differ in priors and computation. These distinctions define the scope of the results throughout the paper.

*Large-language-model tools assisted research planning, code implementation, experiment orchestration, automated audits, and manuscript drafting. Automated replay checks are not independent human verification.

2 Related work and scope

Objective mismatch in model-based control is established prior work. Lambert et al. [1] relate one-step prediction likelihood to control performance and propose reweighting model training. Our intervention changes the planner's objective while keeping the forecaster fixed. The evidence therefore concerns a different part of the control pipeline and does not evaluate their training method.

Terver et al. [2] study the architecture, training objective, and planner of joint-embedding predictive world models. Their Appendix G.3 already examines planning proxies, optimization outside the training action distribution, and local minima. We build on those evaluation questions through same-prefix intervention banks and recorded closed-loop comparisons. Our JEPA comparator consists of their released PushT predictor with locally trained pose readouts; it is not a reproduction of their complete planning pipeline. Diffusion Policy generates controls through conditional denoising [3]. We use a released LeRobot image-policy checkpoint [4] to collect histories, supply proposals, and provide an unchanged policy reference.

Contact-based pushing models connect action-conditioned motion to physical mechanics [5], and ContactNets learns implicit contact geometry and Jacobians [6]. The fitted contact response and known-engine reference used here draw on this established physical approach. Cross-entropy planning and its sample-efficient variants are also established [7]. Our contribution lies in the matched objective comparison, endpoint accounting, and replay evidence. Differences in pretraining, structural priors, and computational budgets prevent an architecture-only comparison between the physical and learned systems.

3 Task, observations, and evaluation

3.1 Observation and intervention protocol

The pinned PushT implementation simulates a position-controlled circular pusher and a rigid T-shaped object. Task coverage is the intersection area between the object and goal divided by goal area. We report unscaled coverage $c \in [0, 1]$ and use the environment's success flag, which requires coverage above 0.95. Episode budgets are at most 300 control steps. Our paper does not substitute a point-position reward for this orientation-sensitive task.

The native development bank contains sixteen policy trajectories, seeds 711000–711015. Four predetermined time fractions are sampled per trajectory without reward or contact filtering, producing 64 contexts. Eight stochastic native Diffusion Policy chunks of length eight are proposed per context and executed from replay-verified identical prefixes. The principal ranking endpoint is step five. There are 17 informative contexts at that endpoint, using the threshold defined below. Training histories use 48 separate trajectories, seeds 710000–710047, with 192 contexts. Native branch training contains 1,536 eight-step branches. The generated-action comparison separately retains uniform and pose-CEM banks of eight twenty-step plans per context.

Learned readouts receive four RGB observations, observed agent positions, recorded controls, and candidate controls. Agent velocity is estimated from observed position differences; simulator velocity is not an inference input. The physical comparator uses the same observation types through a renderer-specific pose estimator, but additionally knows the T geometry, camera scale, and pusher controller. Simulator object states, contact labels, rewards, and future images are used for training labels or retrospective scoring, never as deployed forecaster inputs. These information differences preclude an architecture-only comparison.

3.2 Three endpoints

Let r_{ij} be the true coverage for candidate j at context i, and let $\hat{r}_{ij}$ be its predicted coverage. Predicted poses are converted into coverage by geometric intersection with the goal. The selected index $\hat{j}_i$ first retains candidates within 10^{-8} of maximum predicted coverage, then minimizes fixed pose cost and finally uses candidate order. Let $\Delta_i = \max_j r_{ij} - \min_j r_{ij}$. Informative-state normalized regret is

$$R = \frac{1}{|\mathcal{I}|} \sum_{i \in \mathcal{I}} \frac{\max_j r_{ij} - r_{i\hat{j}_i}}{\Delta_i + 10^{-6}}, \qquad \mathcal{I} = \{i : \Delta_i > 0.01\}. \tag{1}$$

The 10^{-6} denominator stabilizer reproduces the executed metric. All-state mean selected coverage averages $r_{i\hat{j}_i}$ over all 64 contexts. It retains nearly tied contexts that normalized regret excludes. We separately report object-position distance RMSE and wrapped-angle MAE. Episode outcomes are success and maximum coverage attained during the episode; episode-maximum coverage is not terminal coverage or a success probability.

Native development comparator	Position RMSE (pixels)	Angle MAE (radians)	Regret ↓
JEPA pose readout, seed 0	37.983	0.25035	0.17003
JEPA pose readout, seed 1	39.275	0.28551	0.21272
JEPA pose readout, seed 2	37.696	0.28840	0.17284
Damped visual history	9.145	0.05147	0.42196
Fitted contact projection	1.364	0.00788	0.01404

Table 1: Five-step fixed-bank results on 64 inspected development contexts; regret uses the 17 informative contexts. Sensors overlap, but pretraining and physical priors differ.

Matched five-step controller	Successes	Mean peak coverage
First diffusion candidate	3/8	0.69285
Contact-reranked candidate	3/8	0.70020

Table 2: Closed-loop development outcomes. The paired peak-coverage difference is 0.00735, with 95% interval $[-0.01438, 0.03619]$. One paired success is gained and one is lost.

The 64-context comparisons use 20,000 paired bootstrap resamples of whole source trajectories, preserving their four contexts. Physical-versus-readout intervals are adjusted for six comparisons, two deterministic physical arms against three readout seeds. The eight-scene episode comparisons use paired whole-scene bootstrap resampling. Adjustment within one family does not make the full exploratory sequence confirmatory. A single deterministic physical fit compared with three readout seeds is not three independent physical-model replications.

3.3 Comparators

The JEPA reference uses the official initialized PushT predictor with three native-trained pose readouts, selected by development pose loss. The contact comparator estimates the object pose from RGB using a T-shape signed-distance template and known colour/scale priors. Across all 64 native development contexts, current-position RMSE is 0.982 pixels and angle MAE is 0.00510 radians. Three large estimation failures in one training trajectory are retained. The forecaster projects nominal pusher motion against estimated object geometry, using two contact projections per 0.01-second microstep and one rotational-response length fitted from training branches, 59.14682 pixels. A damped visual-history extrapolator supplies a simpler physical reference.

A richer comparator initializes the original collision engine from visual pose histories and observed pusher inputs, using a fresh collision cache and no settling step or replay of the actual future. Both predefined object-velocity initializations, zero and clipped observed pose difference, are retained. It knows simulator geometry, walls, collision laws, and runtime parameters. This is a privileged structural-prior reference, not a learned model.

4 Fixed-bank accuracy does not establish closed-loop gain

Table 1 summarizes the native bank. Contact projection improves all-state selected true coverage over the three JEPA readouts by 0.002024, 0.002319, and 0.002178. Their adjusted trajectory-bootstrap intervals are $[0.000380, 0.004838]$, $[0.000672, 0.004877]$, and $[0.000691, 0.004535]$. These are positive development-bank differences despite small absolute rewards. They establish neither a fair foundation-model comparison nor a closed-loop benefit.

For the corresponding closed-loop comparison, both arms sample eight diffusion-policy chunks at each decision and execute five controls. One takes the first candidate; the other uses contact predictions to rerank the candidates. Both receive the same observation types, proposal count, and execution schedule; reranking adds computation. This first-candidate arm differs from the original policy's eight-action queue. The evaluated scenes, seeds 500000–500007, were already used in development.

The interval in Table 2 includes zero. The sample is too small and selected to support a population-level superiority claim. All sixteen episodes and saved decisions were replayed: RGB frames, chosen indices, predicted poses, success flags, peak coverage, and the paired interval reproduce. This verifies the recorded experiment; because the estimator, predictor, and simulator remain shared, it is not independent validation of contact mechanics.

5 Stronger dynamics priors and generated-action failures

5.1 Known-engine prior

The known-engine reference evaluates 1,024 native-development branches: 64 contexts, eight candidates, and both velocity modes. All eight future prefixes are retained, and decisions are scored at the original fifth step. Position RMSE is 0.941 pixels for zero velocity and 0.957 for observed-difference velocity, compared with 1.364 for contact projection. Yet both modes yield the same mean selected-coverage gain over contact projection, 5.17×10^{-5}. Two contexts improve, none worsen beyond a descriptive 10^{-12} threshold, and 62 have equal coverage. These counts are descriptive, not a test of statistical significance.

The generated-bank comparison includes all 48 training trajectories, 192 contexts, both banks, both velocity modes, and twenty-step forecasts, totaling 6,144 branches. On uniform banks, pooled position RMSE falls from 56.48 under contact projection to 30.38 or 29.04 under the known engine. Mean selected-coverage gains are 0.00359 and 0.00152. On pose-CEM banks, RMSE falls from 35.06 to 12.49 or 13.02; gains are 0.02721 and 0.02317, with coverage losses in eight or eleven contexts. No velocity mode is selected after observing these outcomes.

The last twelve trajectories were previously inspected selection data, not an untouched test set. On their uniform banks, RMSE falls from 56.69 to 29.75 or 26.64, while coverage gains are 0.00573 and -0.00235. Thus the mode with lower positional error can produce worse coverage choices. These endpoints measure different properties; the comparison is not a causal identification of a single missing physical mechanism.

The first independent implementation audit failed on trajectory 710003 with a 1.31-pixel discrepancy. A factorial initialization diagnosis identified shape-reindex ordering. Matching the producer's agent-first reindex semantics, retaining explicit controller microsteps and the original 10^{-6} tolerance, reproduced all forecasts and terminal choices. Maximum state and metric differences are 3.37×10^{-9} and 1.15×10^{-9}. The original failure and post-failure amendment remain public. The native-bank scalar audit agrees to 3.41×10^{-13} in state and 6.20×10^{-14} in metrics. Both checks share the collision engine; neither proves physically correct predictions outside that simulator.

5.2 Generated-action pilot

A separate training-only pilot evaluates all four contexts of trajectory 710000. Random shooting and standard CEM each score 192 twenty-step model rollouts with four piecewise-constant command knots. Their eight highest predicted-coverage plans are then simulated retrospectively; outcomes are not returned to the optimizer.

At the third shooting context, the selected plan is predicted to reach coverage 0.75682 but actually reaches zero. Another retrospectively checked plan reaches 0.71331. The selected plan's position error grows from 0.57 pixels at step five to 145.51 at step ten and 163.24 at step twenty, while nominal agent-position error stays below 10^{-5} pixels. CEM selects a plan reaching 0.91904 in that context, but does not dominate shooting at every context. All 64 checked branches replay identically. Since only the top eight were simulated, their best outcome is not an oracle over all 192 proposals. This pilot demonstrates a specific out-of-bank long-horizon failure; it does not identify its unique cause or establish performance on new scenes.

6 Planning objective as an exploratory control

A second closed-loop family uses 192 twenty-step contact-model rollouts per decision and executes five actions, retaining command bounds, population, iterations, and model. Coverage-first shooting and CEM each succeed on one of the eight development scenes. An exploratory CEM control instead minimizes terminal pose cost

$$d(\hat{p}, \hat{\theta}) = \|\hat{p} - p_{\text{goal}}\|_2 + 20|\operatorname{wrap}(\hat{\theta} - \theta_{\text{goal}})|, \tag{2}$$

with coverage as the secondary tie-breaker. The coefficient has units of pixels per radian and is a fixed heuristic; it is not derived as an optimal task value.

Terminal-pose CEM increases paired peak coverage by 0.21182 versus coverage-first CEM, with bootstrap interval $[0.12826, 0.27928]$. Five successes are gained and none lost; exact two-sided McNemar $p = 0.0625$ does not establish a success-rate difference at level 0.05. Every saved pose-cost plan, episode, RGB observation, and paired statistic reproduces. The result suggests that the planning objective can matter substantially in this scaffold even without a new world model. It remains exploratory because the scenes were already inspected; the positive peak-coverage interval is not unseen success-rate confirmation.

Generated-plan controller	Successes	Mean peak coverage
Coverage-first shooting	1/8	0.72787
Coverage-first CEM	1/8	0.72111
Terminal-pose CEM	6/8	0.93293

Table 3: Exploratory objective control on reused development scenes. The last two rows share the contact predictor and rollout budget. They have a different planning budget from Table 2.

7 Frozen local confirmation: protocol and results

Before accessing the locally reserved episodes, we froze fifty scene seeds, 600000–600049, five controller arms, source/checkpoint hashes, statistical endpoints, and failure handling. All 250 evaluations and their complete decision/environment replays have now finished, retaining fifty episodes per arm. The protocol is locally timestamped and was subsequently published; it is not an externally registered preregistration. No outcome-based stopping, parameter adjustment, checkpoint selection, or scene exclusion was used. The confirmation results below are distinguished from the reused development evidence.

The sole primary comparison is pose-objective CEM against coverage-objective CEM. Both use the identical fitted contact forecaster, 20-step horizon, four five-step control knots, population 64, three iterations, eight elites, standard-deviation floor 10, and five-step receding-horizon execution. Each decision uses 192 model rollouts. The fixed pose cost retains its angular coefficient of 20. The episode budget is 300 control steps, with earlier termination on environment success. The prespecified primary endpoint is paired success-rate difference, tested by the exact two-sided McNemar test at $\alpha = 0.05$. Paired peak-coverage difference receives a percentile interval from 20,000 whole-scene bootstrap resamples. Secondary intervals are descriptive and do not constitute additional confirmatory tests.

The other arms are the released Diffusion Policy using its unchanged native `select_action` queue, and a matched pair drawing eight policy proposals every five executed steps. That pair either takes proposal zero or reranks its eight proposals using the contact forecaster. Sampling rules, budgets, and random-seed schedules are matched; after trajectories diverge, observation-conditioned proposals need not be identical across controllers. Its paired success and peak-coverage differences are secondary analyses. This distinction retains the native eight-action queue as a strong policy reference instead of equating it with the altered proposal-selection interface.

All controllers receive RGB observations and observed pusher positions. Their histories, computation, and structural priors differ: the physical controllers additionally use known object geometry, renderer scale, and controller dynamics. Upstream policy-training exposure to these seeds is unknown. Local developmental non-use therefore does not imply upstream data disjointness, out-of-distribution evaluation, or physical-robot validation. Wall-clock measurements occur under shared machine load and will not support controller-speed claims.

The saved trajectory contains every executed action, RGB frame, observed pusher position, decision record, and simulator state used only for audit. Completed episodes are bound to the frozen protocol by hashes and are not rerun on resume. Every decision and environment step was reconstructed before admission to the final analysis; all five arms begin from identical scene states and pixels, verified from their saved initial records. Pose-estimation failures use the frozen hold-command or proposal-zero fallback and remain counted. Other failures stop the affected worker for investigation. Audit process recovery changes scheduling only, retaining the original failure record. Replay uses the same simulator and forecasting libraries and is not an independent physical model.

7.1 All fifty scenes retained per controller

Table 4 reports all 250 outcomes. Pose minus coverage CEM yields a paired success-rate difference of +0.440, with 24 gained successes and 2 lost successes. The exact two-sided McNemar result is $p = 1.04904 \times 10^{-5}$. The prespecified test supports a success-rate difference favoring pose-objective CEM in this frozen local comparison. It does not establish a new world-model mechanism or a result beyond this task and protocol. The paired mean peak-coverage difference is +0.18187, with descriptive 95% interval $[0.14816, 0.21329]$.

For the secondary matched proposal comparison, reranked minus first-candidate execution yields 14 gained and 6 lost successes, paired success difference +0.160, and mean peak-coverage difference +0.00099, with descriptive 95% interval $[-0.03346, 0.03851]$. This comparison is secondary; the protocol declares no additional confirmatory test. The unchanged native policy is reported as a strong reference, without a prespecified hypothesis test against the CEM arms or equal-compute claim.

Controller	Successes	Mean peak coverage	Mean terminal coverage	Fallbacks
Coverage-objective CEM	11/50	0.76564	0.76134	0
Pose-objective CEM	33/50	0.94751	0.94061	0
Native Diffusion Policy	25/50	0.80010	0.75528	0
First diffusion candidate	25/50	0.88463	0.85604	0
Contact-reranked diffusion	33/50	0.88562	0.86487	0

Table 4: Complete locally frozen confirmation, fifty paired scenes per arm. Success requires coverage above 0.95. Fallbacks count decision events, not failed episodes. The two CEM arms share the forecaster and per-decision rollout budget; policy reference history, priors, execution queues, and computation differ.

All 59,630 executed controls are replayed. Reconstructed plan values and RGB observations match exactly under the numerical checks; maximum simulator-state discrepancy is 0. Exact numerical array equality should not be interpreted as a proof of dtype or byte identity. Independent recomputation from saved trajectories agrees with the counts, paired differences, exact-binomial calculation, and frozen bootstrap draws. Shared forecasting and physics libraries limit the audit to reconstruction of this recorded experiment.

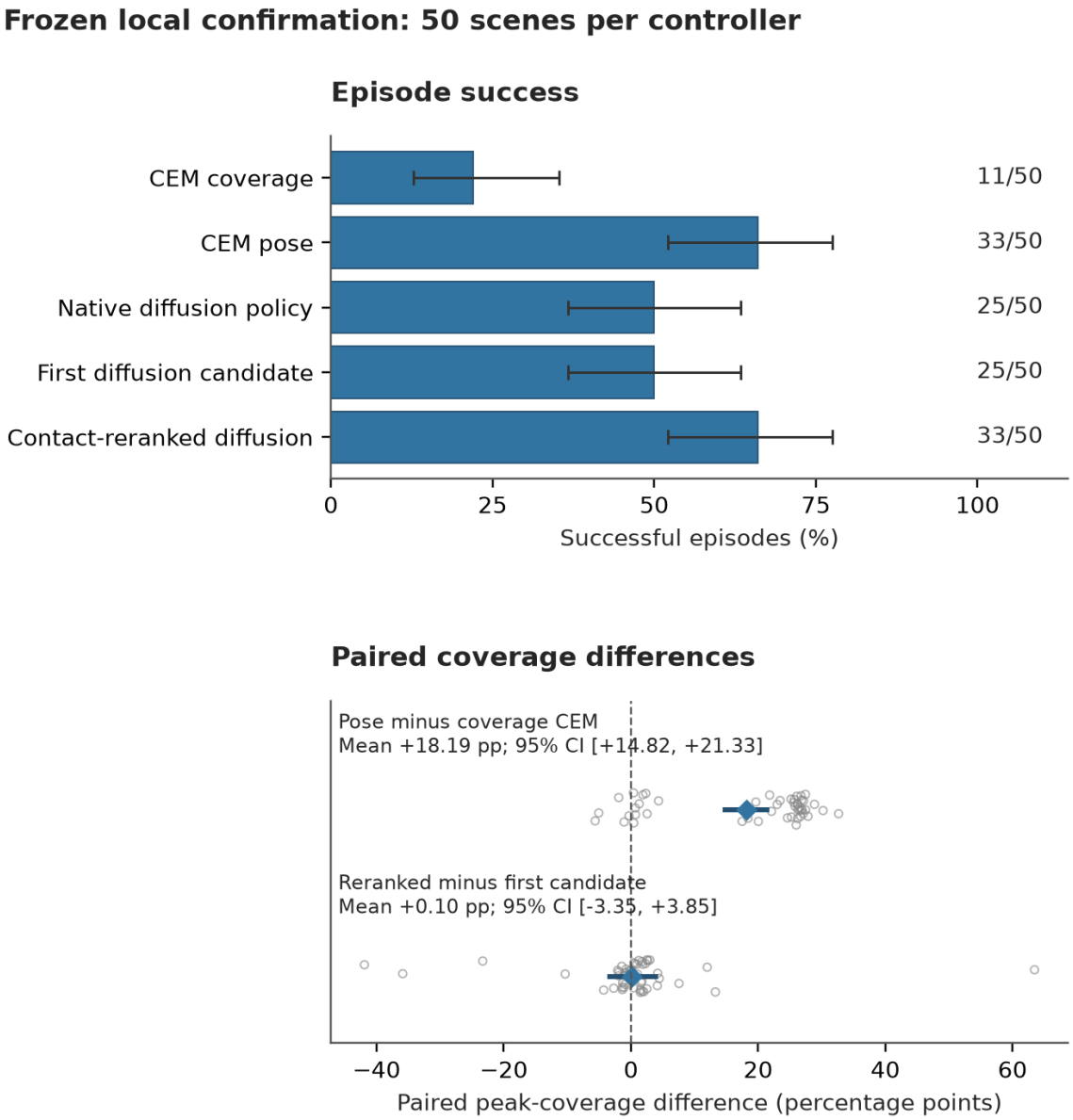


Figure 1: Complete fifty-scene confirmation. Upper bars show success fractions with descriptive marginal Wilson intervals. Lower points show all fifty paired peak-coverage differences per declared comparison, with their mean and scene-bootstrap interval. Marginal bars are not the paired primary significance test.

8 Interpretation and limitations

The primary result is a 44-percentage-point paired success gain from replacing the coverage objective with pose cost in a fixed CEM controller. The forecaster and per-decision search budget are unchanged. This result shows that controller performance in this task depends strongly on how predictions are used. It does not identify why the pose objective works better or establish a new dynamics mechanism. The native policy's 25/50 successes provide a reference, but no CEM-versus-policy hypothesis test or equal-compute comparison was prespecified.

The remaining comparisons concern different endpoints. Contact reranking gains fourteen successes and loses six against first-candidate execution, yet its mean peak-coverage difference is small and its descriptive interval includes zero. Development-bank gains and the known-engine forecasts likewise cannot substitute for episode outcomes. The generated-action pilot demonstrates a specific failure of an optimizer-selected forecast; it neither identifies a unique cause nor estimates the failure rate on new scenes. Positional RMSE alone does not bound every candidate's reward error or the states reached during execution.

All evidence comes from one simulator task. The physical forecasters use renderer-specific pose estimation, known object geometry, and controller priors; the learned systems use external pretraining and different checkpoint selection rules. The fifty confirmation scenes were unused in local development, but upstream training disjointness is unverified. Development scenes were reused, and bootstrap adjustments within individual comparison families do not account for the complete adaptive research sequence. These results do not establish out-of-distribution or physical-robot performance. Replay checks also share the forecasting and collision libraries, so they verify reconstruction of the saved experiment rather than independent contact physics. Wall-clock measurements under shared machine load cannot support speed comparisons.

The archive retains additional predictor, reward, readout, and residual controls that failed their declared gates. A six-model teacher/rollout experiment stopped before all fits completed; its partial weights are backups and are excluded from performance claims. The present confirmation retains all fifty scenes per arm without outcome-based exclusion.

9 Conclusion

With a fixed contact forecaster and per-decision rollout budget, changing the CEM objective raises PushT success from 11/50 to 33/50 in the frozen paired comparison. Prediction accuracy, fixed-bank ranking, and closed-loop control give different conclusions in the supporting experiments. The saved trajectories and complete replay checks make those distinctions inspectable. This is benchmark-specific empirical evidence: evaluation should report the controller's executed outcomes alongside its prediction and action-selection metrics.

Data, code, and use of AI assistance

Code, protocols, source hashes, raw outcomes, failed audits, and reconstruction tools are available at `https://github.com/kineworld/pusht-world-model-research`. The development evidence is pinned to commit `a0994ee6560acdf356ac8efb564351f0fdb34f17`; the confirmation data and completed audit are pinned to `59d166f961b0e619b313e6edc1c54c05aeff92fa`. The confirmation supplement contains all 250 linked trajectory, decision, and replay records with numerical recomputation tools. It does not include third-party pretrained weights or the complete external simulator installation. AI assistance is disclosed on the first page; automated reconstruction is not independent human peer review. The author is responsible for the manuscript and its evidence.

A Evidence map and reconstruction scope

The following paths are relative to `decision_rank/` in the pinned archive. Each result JSON records executed-source and protocol hashes; its paired audit records the checked result hash. Reconstruction coverage is explicitly limited to the saved experiment.

- Frozen fifty-scene confirmation: `journal_confirmation_result/validation.json`, `independent_audit.json`, and all 250 linked trajectory/decision/audit records in the accompanying supplement. Every saved plan and environment step reconstructs; paired initial conditions and raw statistical arithmetic are checked. This evidence belongs to the current confirmation revision, separately from the older pinned development archive.

- Fixed native ranking: `visual_contact_projection_result/validation.json` and `independent_audit.json`. Forecasts and interval grouping reproduce; the physical predictor remains shared.
- Matched policy reranking: `visual_contact_closed_loop_result/validation.json` and its audit. All sixteen episodes and saved decisions replay.
- Generated-action pilot: `contact_planner_pilot_result/validation.json` and its audit. All selected top-eight branches and optimizer decisions reconstruct; unselected plans have no ground-truth outcomes.
- Objective control: `contact_search_closed_loop_result/validation.json`, `contact_pose_objective_result/validation.json`, and their audits. Saved observation-conditioned plans and paired coverage statistics reproduce.
- Richer physical reference: `contact_engine_train48_v2_result/summary.json`, `reindex_independent_audit.json`, and `failed_audit_record.json`; `contact_engine_native_result/summary.json` and its audit. Shared-engine forecasts, terminal scores, selected indices, and separate group arithmetic are checked.

The archived chunk-restoration tool reconstructs large self-owned artifacts from recorded chunks. Dependency, checkpoint, and simulator provenance should be inspected before rerunning a producer. The old reading-copy PDF and presentation decks in the archive are historical documents and do not represent this manuscript.

References

[1] N. Lambert, B. Amos, O. Yadan, and R. Calandra. Objective Mismatch in Model-based Reinforcement Learning. Proceedings of the 2nd Conference on Learning for Dynamics and Control, PMLR 120:761–770, 2020. https://proceedings.mlr.press/v120/lambert20a.html.

[2] B. Terver, T.-Y. Yang, J. Ponce, A. Bardes, and Y. LeCun. What Drives Success in Physical Planning with Joint-Embedding Predictive World Models? arXiv:2512.24497, 2025. https://arxiv.org/abs/2512.24497.

[3] C. Chi et al. Diffusion Policy: Visuomotor Policy Learning via Action Diffusion. arXiv:2303.04137, 2023. https://arxiv.org/abs/2303.04137.

[4] LeRobot contributors. Diffusion PushT pretrained image-policy checkpoint and model card. https://huggingface.co/lerobot/diffusion_pusht. Accessed 5 October 2026.

[5] J. Zhou, J. A. Bagnell, and M. T. Mason. A Fast Stochastic Contact Model for Planar Pushing and Grasping: Theory and Experimental Validation. Robotics: Science and Systems, 2017. https://arxiv.org/abs/1705.10664.

[6] S. Pfrommer, M. Halm, and M. Posa. ContactNets: Learning Discontinuous Contact Dynamics with Smooth, Implicit Representations. Conference on Robot Learning, 2020. https://arxiv.org/abs/2009.11193.

[7] C. Pinneri et al. Sample-efficient Cross-Entropy Method for Real-time Planning. Conference on Robot Learning, 2020. https://arxiv.org/abs/2008.06389.