PRP: DIAGNOSING STATE READABILITY BEYOND PREDICTION ERROR IN JEPA WORLD MODELS

Anonymous authors
Paper under double-blind review

ABSTRACT

World models predict how an environment evolves from observations and actions. When these predictions are learned representations, low prediction error leaves open which physical states remain readable and whether the rollout adds useful state beyond the initial observation. We introduce *Paired Rollout Probing (PRP)*, an evaluation protocol comparing observed, predicted, and frozen initial representations through a shared readout. Our contributions are threefold: a quantity-specific diagnosis of readout transfer, a calibration test for recoverable readout mismatch, and persistence-controlled evaluation of observation refreshes and adaptation. Position remains highly readable over 15 predictions in two Meta-World-trained models. In a DROID-trained V-JEPA 2-AC case study on Meta-World, position readability drops sharply within two predictions, but training-only mean/scale calibration restores much of the early gap. A held-out matched restart improves next-step object decoding by 26.2 percentage points. Adaptation lowers one-step latent error from 0.94 to 0.40, while gains over persistence depend on the state quantity, objective, and horizon. These comparisons distinguish readout compatibility, recovery, and added state value, enabling more specific assessments of latent prediction than aggregate error alone.

1 INTRODUCTION

Latent world models predict future representations for planning and control (Assran et al., 2025; Ha & Schmidhuber, 2018; Hafner et al., 2019; Hansen et al., 2024; Zhou et al., 2024). Latent error measures aggregate representation discrepancy; task success measures the combined model, planner, and interaction loop. State-specific readouts connect these evaluations to physical quantities such as hand and object position, clarifying what a lower prediction error means for an imagined future.

We introduce *Paired Rollout Probing (PRP)*, which applies a fixed observation-trained probe to three representation paths on the same trajectories (Figure 1). Probe accuracy defines *state readability*. Encoded observations establish the readout reference, predictions test temporal transfer, and the frozen initial latent measures persistence. Calibration and matched restarts then test how feature alignment and fresh observations change these comparisons.

Readability decay identifies a transfer problem. We define *readability decay* as decreasing accuracy of a fixed observation-trained probe along predicted trajectories. PRP compares this decline with observations and initialization. In the shifted case, end-effector accuracy falls from 0.93 to 0.64 and object accuracy from 0.89 to 0.65 between predictions one and two. Two native models instead sustain high position readability (Figure 3).

Calibration tests recoverable mismatch. A fixed probe expects features in the observation coordinate system. Matching predicted feature means and scales to training observations restores much of the early gap on held-out trajectories (Figure 5). This separates poor direct transfer from state accessible after a simple correction, motivating a readout-alignment test before attributing low scores to state loss.

Persistence comparisons measure added state value. We define *added state value* as the signed decoding gain over a specified initialization baseline at a given horizon. Matched restarts measure refresh duration; adaptation tests gains over the initial label or latent. A held-out observation restart

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.

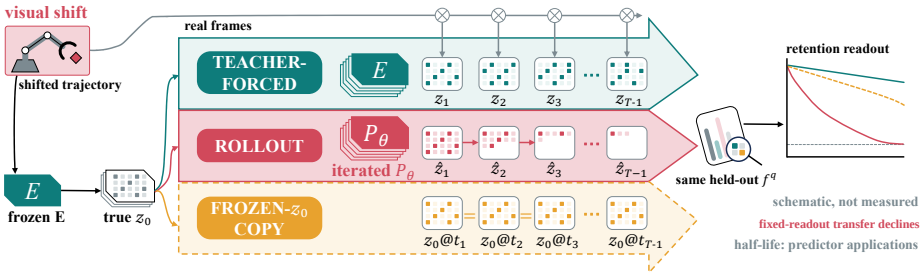


Figure 1: **Three latent paths, one readout.** The observation-trained probe evaluates encoded observations (the teacher-forced branch), predictions, and frozen initial latents on held-out trajectories. Observation and persistence comparisons measure transfer and added value, respectively. The right panel is schematic.

adds 26.2 points to next-step object decoding, while adaptation gains depend on quantity and horizon. Action ranking separately evaluates the latent cost.

Contributions. PRP provides (i) paired observation, rollout, and persistence readouts for locating quantity-specific transfer failure; (ii) a training-only calibration test showing recoverable readout mismatch in the shifted case; and (iii) matched refresh and adaptation comparisons that quantify gains over time and initialization. High native-distribution readability is replicated across two models. Calibration, restarts, and adaptation are detailed analyses of one shifted checkpoint. Together, these results identify when to improve readout alignment, when to evaluate observation refreshes, and when a lower prediction loss still needs a state-specific baseline comparison.

2 Related Work

From successful planning to identifiable state. Latent world models support planning and learned behavior (Ha & Schmidhuber, 2018; Hafner et al., 2019; 2023; Hansen et al., 2024). Predictive features support reward-free planning (Assran et al., 2025; Sobal et al., 2025; Zhou et al., 2024), while V-JEPA 2.1 improves dense features (Mur-Labadia et al., 2026). JEPA-WMs compares modeling and planning choices and evaluates rollout state error using an observation-trained decoder (Terver et al., 2026). PRP builds on state decoding by pairing observation, prediction, and persistence paths. These comparisons separate readout transfer from gains over initialization; calibration tests recoverable mismatch, and matched restarts measure how long fresh observations help.

From compounding error to quantity-specific decay. Regularization, training objectives, and short rollouts address multi-step error (Janner et al., 2019; Talvitie, 2014; Venkatraman et al., 2015). Exposure-bias studies examine training–generation mismatch (Arora et al., 2022; Bengio et al., 2015; Lamb et al., 2016; Ranzato et al., 2016); imitation learning addresses visited-state shifts (Ross et al., 2011). PRP adds quantity- and horizon-specific observation and persistence comparisons, locating transfer failure and the value of advancing the representation.

From encoded information to a transferable readout. Linear probes measure information accessible to a chosen readout (Alain & Bengio, 2017; Belinkov, 2022; Hewitt & Liang, 2019). Video-model analyses study physical representations across depth and geometry (Joseph et al., 2026; Punzo et al., 2026), concept interventions (Alam, 2026), and discrete symbols in frozen features (Liu, 2026). PRP tests temporal compatibility: whether the same observation-trained feature transformation and classifier recover state after repeated prediction. Accuracy measures compatibility with an existing readout. Our calibration comparison additionally asks how much of its failure can be recovered by aligning feature statistics while keeping the classifier fixed.

From adaptation gains to gains over persistence. Test-time adaptation uses unlabeled inputs (Sun et al., 2020; Wang et al., 2021); AdaJEPA adapts dynamics inside control (Wang et al., 2026), and hierarchical models use temporal abstraction (Zhang et al., 2026). PRP complements these objectives with state-specific gains over initialization and action ranking. Appendix E compares evaluation questions, including video-rollout evaluation (Huang et al., 2026).

3 PAIRED ROLLOUT PROBING

PRP uses a shared readout and evaluation mask across latent paths within each model and condition. Separate probes serve each model, quantity, and corruption condition. An encoder E maps frame x_t to $z_t = E(x_t) \in \mathbb{R}^{K \times d}$, with K spatial tokens of dimension d. The action-conditioned predictor consumes latent history, actions a_t, and proprioceptive states s_t:

$$\hat{z}_{t+1} = P_\theta([z_0, \hat{z}_{1:t}], a_{0:t}, s_{0:t})[-K:], \quad t = 0, \ldots, T-2. \tag{1}$$

Only the initial visual latent comes from an observation. Subsequent visual latents are predicted, with actions and proprioception from the recorded trajectory. Native models use their released rollout interfaces.

Fixed readouts and encoder reference. For each quantity q, spatial grid pooling and PCA form a feature map g, and a linear classifier f^q is trained on encoded observations from training trajectories. Quantities are binned end-effector and object position, gripper state, contact, and, for the in-distribution models, task identity. On held-out trajectories we compute

$$\mathrm{enc}^q(t) = \mathrm{bAcc}(y_t^q, f^q(g(z_t))), \qquad \mathrm{roll}^q(t) = \mathrm{bAcc}(y_t^q, f^q(g(\hat{z}_t))). \tag{2}$$

Here bAcc is balanced accuracy, the mean recall across retained classes. Chance is $1/C_{q,t}$ for $C_{q,t}$ retained classes. A high reference score establishes that the quantity is available in the observation representation and readable by the fitted probe. The rollout score then measures transfer of that same readout.

Figure 2 shows the spatial readout. Grid pooling concatenates cell averages in a fixed order, preserving coarse layout. Global mean pooling averages across the grid; individual token features may still encode position. The same fitted transformation is applied across latent paths.

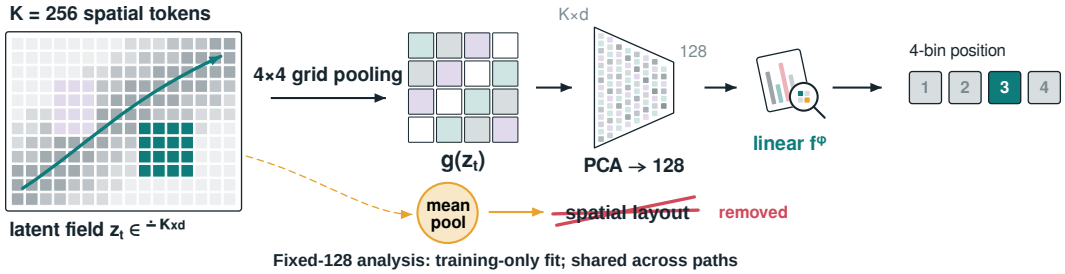


Figure 2: **Spatial readout construction.** Grid pooling preserves the spatial arrangement of pooled cells before PCA and linear classification. The mean-pooling branch removes the explicit grid arrangement; token features may still encode position. The fitted transformations are held fixed across the three paths. The fixed-128 calibration analysis fits all transformations on training trajectories; dimension-selection details for the initial surveys are given in Section 4.

Persistence baselines. The label-persistence baseline is $B^q_{\mathrm{label}}(t) = \mathrm{bAcc}(y_t^q, y_0^q)$. It repeats the initial true label. The frozen-latent baseline is $B^q_{\mathrm{latent}}(t) = \mathrm{bAcc}(y_t^q, f^q(g(z_0)))$. It repeats the initial encoded observation through the same probe. These baselines answer different questions: how much the labels have changed, and how useful the initial representation remains.

We report signed differences $\Delta^q_j(t) = \mathrm{roll}^q(t) - B^q_j(t)$ for each baseline j. Positive differences favor rollout decoding; negative differences favor persistence. We report paired point estimates at each horizon. Their signs locate gains and deficits within the trajectory rather than establishing statistical significance.

Decay and displacement. Decoding half-life is the first interpolated horizon at which the rollout score reaches halfway between its step-1 value and chance, indexed from the initial observation at $t = 0$. We measure relative latent displacement with the copy-index

$$\mathrm{CI} = \mathbb{E}\left[\frac{d_0}{d_0 + d_k}\right], \quad d_0 = \|\hat{z}_k - z_0\|_1, \quad d_k = \|\hat{z}_k - z_k\|_1. \tag{3}$$

For a nonzero denominator, an exact initial-latent copy scores zero, an exact future latent scores one, and equal distances score one half. The index describes relative displacement alongside absolute error and probe scores.

4 EXPERIMENTAL DESIGN

The experiments test readout transfer, recovery through calibration, and added value over initialization. Two native models assess sustained transfer; one shifted checkpoint supports detailed interventions. Noise and stride comparisons examine sensitivity within existing evaluations.

Models and distributions. The shifted case is DROID-trained V-JEPA 2-AC evaluated on Meta-World renders (Assran et al., 2025; Khazatsky et al., 2024; Yu et al., 2020), with a frozen ViT-g encoder ($K = 256$, $d = 1408$) and a 305M predictor. Simulator-state labels support a quantity-specific diagnosis under this real-to-sim transfer.

The native comparisons are Meta-World-trained JEPA-WM (17.6M adaptive-layer-normalization predictor) and DINO-WM (21.3M predictor with a separate action encoder), both using frozen DINOv2-S encoders (Oquab et al., 2023; Terver et al., 2026; Zhou et al., 2024). They test whether readout transfer persists in two native-distribution predictors. Training, capacity, and design vary jointly; the fixed-model corruption sweep isolates input sensitivity.

Trajectories and readouts. The shifted case uses 240 scripted-expert Meta-World trajectories, split 120/120 by trajectory using a seed-0 permutation. Native-distribution analyses use 240 windows sampled from the released Meta-World training data, with a separate 120/120 probe split. They assess readability within the model's training distribution. Standard windows contain 16 frames and 15 prediction transitions. A 4×4 spatial grid is pooled and concatenated, standardized, projected by PCA, and classified by balanced multinomial logistic regression. Transformations and classifiers are fitted to training-half observations pooled across time. The initial surveys select PCA dimension with a sweep that includes evaluation trajectories; the calibration and adaptation analyses fix it at 128. Stride 6 selects 256. At each quantity and horizon, classes with fewer than five evaluation samples are excluded, and fewer than two surviving classes yields no score. Paired paths share the same mask.

Why position is the main diagnostic. Contact and gripper have weak observation readouts in the shifted checkpoint, whereas position is strongly readable. Position thus supports a clear test of prediction-side deterioration. Weak DROID observation probes (Appendix C) separately identify a boundary of this global readout design.

Shift strength and temporal stride. Gaussian pixel noise at uint8 levels $L \in \{0, 2, 4, 6, 8, 16, 32, 64\}$ perturbs the native JEPA-WM inputs. Each positive level uses three perturbation seeds on fixed trajectories; level zero has one unperturbed run. Probes are refitted for each level and seed. The sweep tests rollout transfer when corrupted observations remain readable. Separately, stride 6 increases the interval between sampled frames in the shifted case. Actions are differences between sampled end-effector states and proprioception is the sampled current state. The stride comparison therefore changes physical time, conditioning magnitudes, and the selected PCA dimension.

Training-only feature calibration. In a common observation-fitted PCA-128 space, let u_t and $\hat{u}_t$ denote observed and predicted features. At each horizon, training trajectories provide featurewise means and standard deviations. We transform evaluation predictions as

$$\hat{u}_t^{\mathrm{cal}} = (\hat{u}_t - \mu_t^{\mathrm{roll}}) \odot \frac{\sigma_t^{\mathrm{obs}} + \epsilon}{\sigma_t^{\mathrm{roll}} + \epsilon} + \mu_t^{\mathrm{obs}}. \tag{4}$$

The observation-trained classifier remains fixed. This diagonal affine correction uses paired training observations and predictions at each horizon, with no evaluation labels. A full affine ridge map provides a more flexible comparison. Calibration tests recoverability under this data access; persistence and action ranking test its usefulness. A separate continuous end-effector readout uses ridge regression and R^2 to examine movement within position bins.

Restart controls. At $t_0 = 8$, observed-, predicted-, and donor-latent restarts use z_{t_0}, $\hat{z}_{t_0}$, and another trajectory's observation, respectively. All use truncated history and the same subsequent actions and proprioception. Their labels are A, C, and B; base denotes uninterrupted prediction. A–C measures the effect of replacing the predicted latent with the current observation under matched history. The main single-refresh analysis uses a held-out PCA-128 probe. The periodic experiment uses a separate probe fitted and evaluated on all 240 trajectories; it describes within-sample refresh timing. We keep these protocols separate.

Adaptation and action ranking. The encoder is frozen and 98 predictor layer-normalization tensors are updated with Adam at 10^{-4}. One-step adaptation uses 1800 training-half transitions; transductive adaptation continues for 1800 evaluation-input transitions without physical-state labels. Four-step adaptation uses 480 updates, from 120 training trajectories and starting steps $0, 3, 6, 9$. All conditions are evaluated on the same 120 evaluation trajectories. Offline ranking compares future-latent distances under recorded actions and another trajectory's actions, without magnitude matching. The initial two-stride survey uses all 240 trajectories; calibration and adaptation comparisons use the evaluation half. We report the fraction of pairs for which the recorded action has lower cost.

5 Results

5.1 Readability decay depends on the model–data configuration

Position remains highly readable across 15 native-distribution predictions, while the shifted checkpoint shows a sharp early decline. Figure 3 pairs the readouts with their observation references, then tests input noise and temporal sampling. The first comparison establishes where transfer succeeds or fails; the latter comparisons examine sensitivity within an existing model.

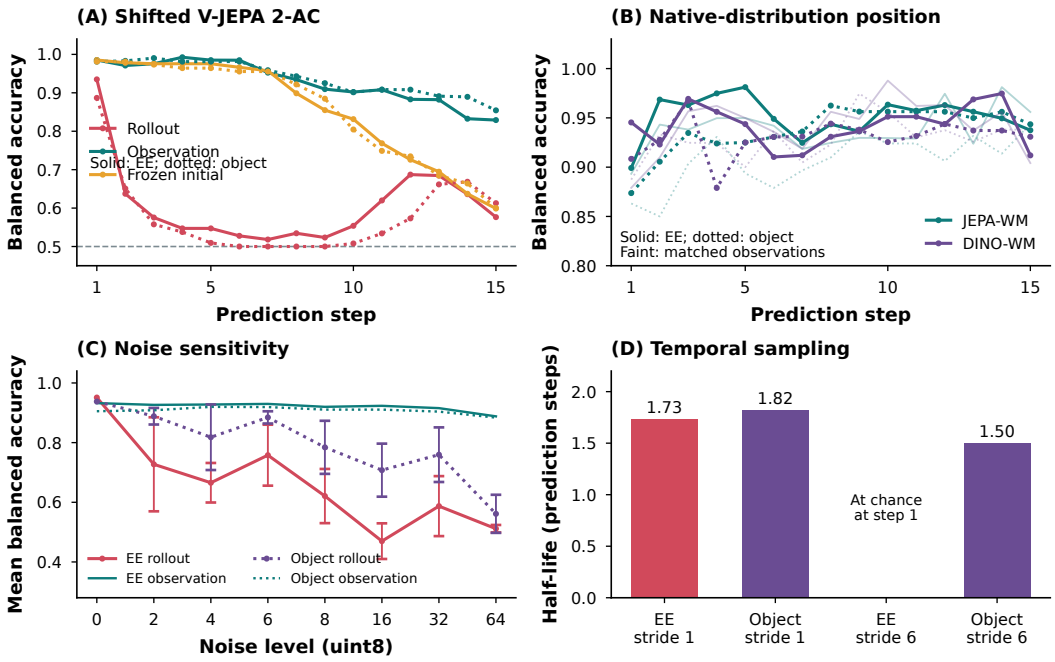


Figure 3: **Position-readout transfer and its conditions.** (A) Shifted-model observation, rollout, and frozen-initial curves; solid lines denote end-effector position and dotted lines object position. (B) Native-model rollout curves and their faint, horizon-matched observation references. (C) Mean decoding over steps 1–15; error bars are population SD across three perturbation seeds at positive noise levels, with one run at level zero. (D) Half-lives interpolated from full-precision scores; stride-6 EE is already at chance. Points are stored measurements, and lines connect them. The 0.5 reference in A applies to the shifted position task; native class definitions differ.

Advancing the latent can reduce direct access to position. In the shifted checkpoint, first- and second-step accuracies are 0.93/0.64 for the end effector and 0.89/0.65 for the object. Both frozen-initial readouts round to 0.98 at step two. Paired bootstrap intervals for frozen-minus-rollout differences exclude zero at that step: 0.34 [0.27, 0.40] for the end effector and 0.33 [0.27, 0.40] for the object. Thus the early predictions are less useful to this fixed readout than the retained initial representation. Half-lives are 1.73 and 1.82 prediction steps, respectively, using the first predicted score and chance 0.5 to define the crossing threshold.

Repeated prediction can also sustain readout transfer. JEPA-WM and DINO-WM maintain high position decoding across their native-distribution windows. Their observation curves in Figure 3B provide quantity- and horizon-matched comparisons. The JEPA-WM 20-frame extension retains end-effector accuracy of 0.90–0.99 and object accuracy of 0.87–0.96 over 19 predictions. Gripper and contact ranges are 0.55–0.86 and 0.70–0.90; task identity remains 1.00. Different state quantities therefore have different readout profiles, while position transfer remains strong in this extension. The cross-model comparison establishes configuration dependence; the models differ jointly in training data and architecture.

Readable inputs can yield less readable predictions. With the native JEPA-WM held fixed, mean end-effector rollout accuracy decreases from 0.95 at zero noise to 0.51 at level 64, and object accuracy from 0.94 to 0.56. Corresponding observation means remain about 0.88–0.94. The response is nonmonotonic: end-effector means at levels 4, 6, and 8 are 0.67, 0.76, and 0.62. The seed variation in Figure 3C is consequently part of the result. This separates sensitivity of prediction-side readout transfer from the readability of the perturbed observations.

Object half-life is 1.50 predictions at stride 6 versus 1.82 at stride 1. End-effector decoding at stride 6 is at chance from the first prediction, so that run supplies no estimable half-life. This is a quantity-specific temporal-sampling comparison. Its simultaneous changes to conditioning and PCA dimension make it a boundary test of the observed pattern.

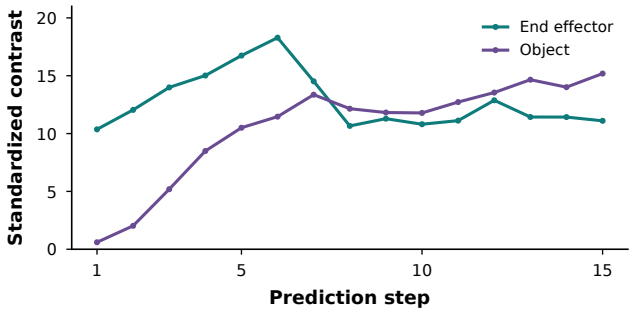


Figure 4: **Native JEPA-WM geometry.** Standardized probe-minus-random variance contrasts relative to norm-matched isotropic perturbations. The denominator is the SD of 20 null draws. These quantify subspace spread; Appendix B defines the asymmetric random-subspace sampling.

Geometry complements the readout score. In native JEPA-WM, projected variance relative to observed features ranges from 0.91–1.01 in end-effector probe directions and 0.80–1.07 in object directions, versus 0.79–0.86 in random subspaces. Figure 4 compares these contrasts with isotropic errors of the same per-sample norm. The standardized contrast is 10.4–18.3 for the end effector; object values rise from 0.60 at step one to 5.2–15.2 from step three. These second-moment measurements characterize how prediction error is distributed across directions. The next analysis tests the more direct question of whether aligning predicted features restores label recovery in the shifted checkpoint.

5.2 Much of the early readout mismatch is recoverable

Training-only mean/scale calibration restores much of the early position-readout gap without changing the observation-trained classifier. Figure 5 compares raw transfer, diagonal calibration, and a full affine map within the same PCA-128 analysis. At step two, end-effector accuracy rises from 0.64 to 0.97 and object accuracy from 0.66 to 0.97. Their observation references are 0.97 and 0.98. The raw object score differs slightly from the initial survey because this analysis refits its own common PCA and probe; all calibration contrasts use that shared fit.

Early readability decay is therefore substantially recoverable through affine feature alignment. The relative performance of full and diagonal correction varies across horizons. This diagnosis uses horizon-specific statistics estimated from paired training observations and predictions.

Recovering transfer and exceeding initialization are distinct. Averaged over the full 16-point window including initialization, the frozen-minus-rollout gap decreases from 0.234 to 0.029 for the end effector and from 0.255 to -0.002 for the object. The latter is a small average point estimate in favor of calibration; horizon-wise object gap intervals include zero throughout. End-effector gaps remain positive with intervals excluding zero at several intermediate horizons. Calibration therefore

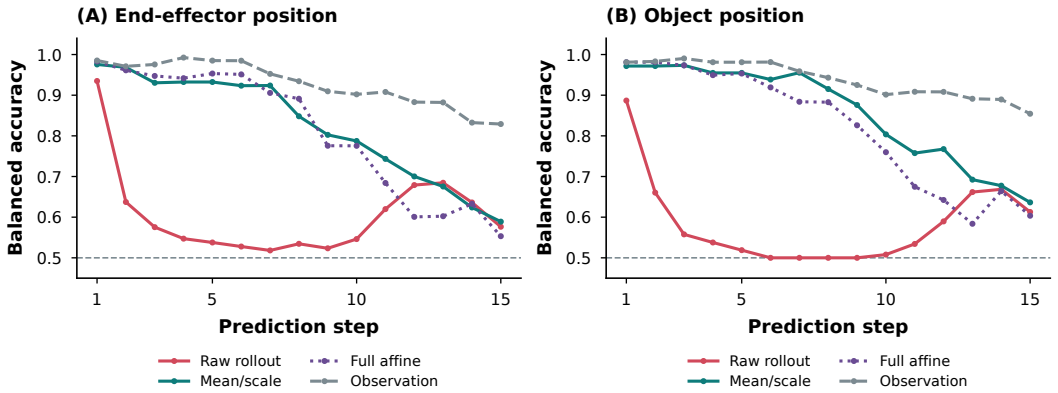


Figure 5: **Recoverable readout mismatch in the shifted checkpoint.** The same observation-trained classifier reads raw and corrected predicted features. Mean/scale calibration uses training-half statistics separately at each horizon; the full affine map uses ridge regression on paired training features. Compare each corrected curve with raw transfer and the observation reference within its panel. Both quantities show large early recovery. Curves are point estimates on 120 evaluation trajectories.

repairs a substantial readout mismatch, while the initial observation remains a competitive state reference.

The continuous end-effector analysis clarifies why persistence is demanding. At step four, the observation-trained rollout readout has $R^2 = 0.186$, a rollout-trained readout reaches 0.920, and initial-state persistence reaches 0.980. At step eight these values are 0.089, 0.638, and 0.724, with frozen-latent regression reaching 0.780. Continuous targets expose movement within the coarse position bins while retaining the same conclusion: readout recovery and added current-state value require separate comparisons.

5.3 Observation refreshes provide a brief matched-readout benefit

A fresh observation improves the next predicted state, and its advantage diminishes with further prediction. The held-out restart comparison in Figure 6A isolates the replacement latent under matched truncated history. Observed and predicted restarts score 0.770 and 0.508 at step nine, a 26.2-percentage-point contrast. The donor restart scores 0.536 and uninterrupted rollout 0.500. The current trajectory's observation provides the largest immediate gain.

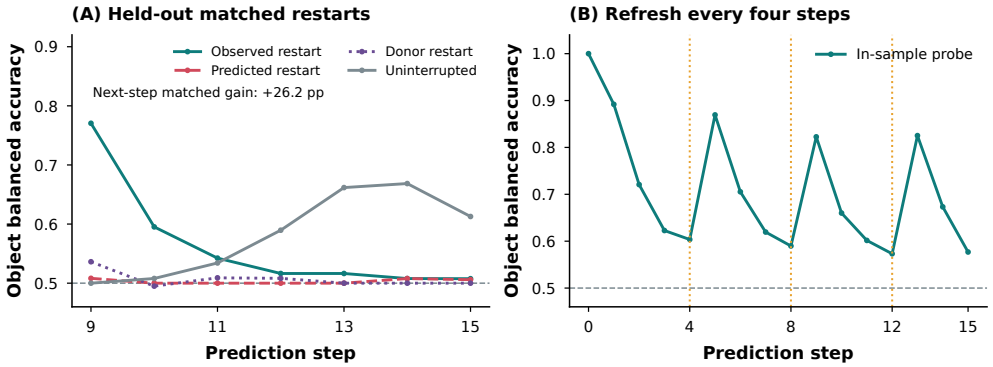


Figure 6: **Fresh observations improve immediate readout.** (A) Held-out object decoding after restart at step eight, using a common observation-trained probe. Observed-minus-predicted gain falls from 0.262 at step nine to 0.095 at step ten and 0.042 at step eleven. (B) A separate in-sample probe experiment shows repeated responses after refreshes at steps 4, 8, and 12. Its next-step peaks are 0.869, 0.823, and 0.825. The panels use different probe-fitting protocols. Lines connect measured points.

The response trajectory matters as much as its peak. The matched gain falls to 9.5 points at the second prediction after restart and 4.2 at the third. By step 14, uninterrupted prediction reads 0.669 versus 0.508 after an observed restart. A refresh can therefore improve immediate access while its truncated-history path has weaker later decoding. The in-sample periodic experiment exhibits repeated peaks followed by decline between refreshes. Between-refresh decoding reaches 0.577 at step 15. Together, the results motivate evaluating both the immediate benefit and its duration under the actual restart rule.

Calibration changes readout coordinates; refresh changes the predicted trajectory. The restart contrasts use raw readouts. A calibrated comparison would require statistics estimated separately for each restart arm, since calibration fitted to the uninterrupted rollout specifically favors that path.

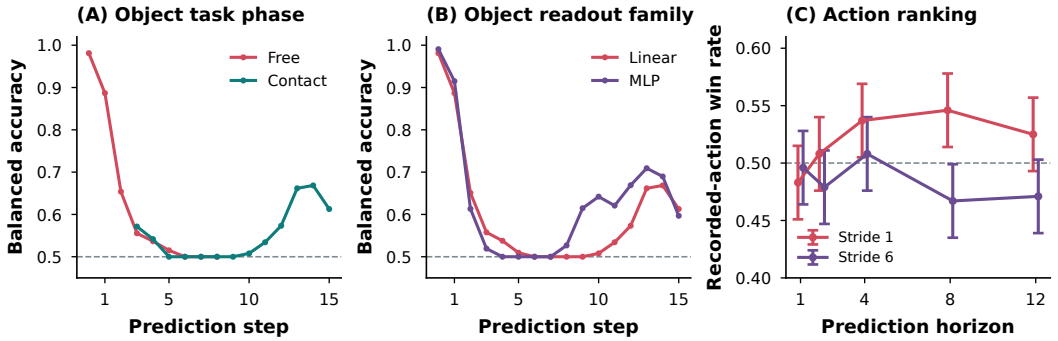


Figure 7: **Task phase, readout family, and action ranking.** (A) Object decoding stratified by contact, with at least 15 samples per reported stratum. Missing strata are left unconnected. (B) Observation-trained linear and MLP object readouts on the shifted trajectories. (C) Recorded-versus-donor action ranking at two strides, using token-level ℓ_1 cost on 240 trajectories. Error bars are SE, computed as population SD of wins divided by $\sqrt{240}$. Donors are other trajectories' actions.

Recovery depends on phase and readout. Contact-stratified object decoding associates the later rise with contact (Figure 7A). An observation-trained MLP also deteriorates early, with a different later recovery. A horizon average therefore conceals both the weakest transfer interval and its phase-dependent recovery.

5.4 Lower prediction error yields selective gains over persistence

Adaptation strongly improves decoding relative to the unadapted rollout. Its added value over initialization depends on the quantity and objective. Figure 8 evaluates end-effector and object position separately against label persistence and frozen latents. One-step latent error decreases from 0.943 to 0.400 with inductive adaptation and to 0.362 after transductive continuation. Four-step adaptation yields 0.564 under the same one-step evaluation metric.

Object and end-effector readouts give different answers. Inductive one-step adaptation remains at or below frozen-latent decoding over steps 2–8. Transductive end-effector decoding slightly exceeds label persistence at steps 3–5 and ties frozen-latent decoding there. Four-step adaptation improves object decoding over frozen latents throughout steps 2–8, with a minimum difference of 0.002, and exceeds label persistence at steps 5–8. Its end-effector readout retains negative differences to both baselines. Reporting the baselines separately identifies which aspect of initialization an update improves on. The worst end-effector differences to frozen latents over steps 2–8 are -0.020 for one-step and -0.047 for four-step inductive adaptation, versus -0.009 and $+0.002$ for object position (Table 1). Thus the same objective changes state access differently across quantities.

Action-ranking gains depend on the cost. The initial two-stride action survey spans win rates of 0.467–0.546 (Figure 7C). A separate held-out comparison fixes candidate pairs across three costs. At horizon eight, raw token ℓ_1 scores 0.633 [0.550, 0.717], PCA-space ℓ_2 scores 0.542 [0.450, 0.633], and calibrated PCA-space ℓ_2 scores 0.392 [0.308, 0.483]. These are pointwise 95% bootstrap intervals on 120 trajectories. The raw cost favors recorded actions at this horizon, while the calibrated

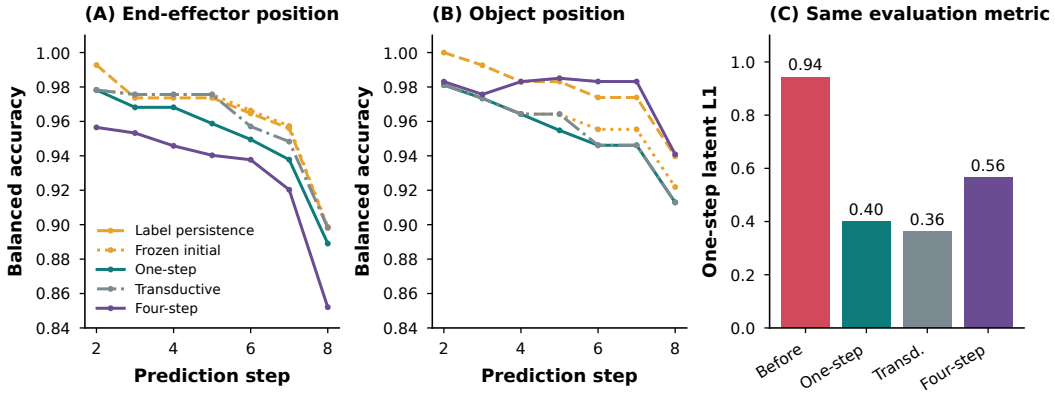


Figure 8: **Adaptation gains depend on the state baseline.** (A,B) Readout scores at steps 2–8 for inductive one-step, transductive one-step, and inductive four-step adaptation. Compare each quantity with its own label and frozen-latent baselines. (C) One-step latent ℓ_1 error evaluated on the same 120 trajectories for every condition, including the model trained with the four-step objective. All curves and bars are point estimates from individual adaptation runs.

cost favors donors. The same calibration that improves state readability therefore has a different effect on this action-ranking criterion.

After inductive one-step adaptation, the horizon-eight ranking point estimate is 0.592, versus 0.558 before adaptation; four-step adaptation yields 0.500. These single-run scores complement the readout curves without establishing a reliable improvement in action selection. Candidate construction and cost definition are part of the evaluated behavior. AdaJEPA studies adaptation inside an executed control loop (Wang et al., 2026); our comparisons examine the information supplied by fixed recorded trajectories.

6 Discussion and Conclusion

PRP separates readout compatibility, recovery after calibration, and state value beyond initialization. Raw probe scores can recover after alignment while remaining close to persistence; a cost built from corrected features can also rank actions less successfully.

What the diagnosis changes. First establish a strong observation readout. If transfer then deteriorates, test training-only calibration and compare recovered scores with both persistence baselines. For refreshes, match restart history and inspect the full response. For action selection, evaluate the actual latent cost. Each comparison identifies the specific failure that a model or readout change should address.

Evidence scope. Native-distribution position readability is replicated across two models; the calibration, refresh, and adaptation findings concern one shifted V-JEPA 2-AC checkpoint. Native windows come from the released training distribution. The initial survey's PCA dimension selection includes evaluation trajectories, whereas the fixed-128 calibration analysis uses training-only fits. Adaptation results are individual runs; bootstrap intervals condition on fitted readouts, and action tests are offline. The periodic refresh experiment is in-sample. These distinctions determine which comparisons establish held-out readout recovery and which describe sensitivity within the observed trajectories.

Conclusion. PRP makes latent prediction assessable in terms of physical-state access and added value. Two native-distribution models sustain position readability, while the shifted case exhibits rapid decay that is substantially recoverable through feature calibration. Matched observation refreshes provide a brief decoding benefit, and adaptation produces selective gains over persistence despite broad improvements in prediction error. Evaluating these properties separately reveals what a predicted future contributes beyond an accurate latent loss or a readable initial observation.

Reproducibility statement

Sections 3 and 4 specify readouts, trajectory splits, calibration, restart controls, and adaptation. Appendix B gives statistical and implementation details, and Appendix D reports numerical comparisons. The source package includes result files, the evaluation scripts supplied with them, and figure-generation code that reads those results directly. Bootstrap and seed uncertainty are identified in the relevant captions. Reproducing model inference additionally requires the released checkpoints and trajectory or latent caches described by the evaluation code.

Ethics statement

This study analyzes public world-model checkpoints, simulated manipulation trajectories, and an existing robot dataset. The evaluation concerns model behavior under the specified offline conditions. Dataset and model use should follow release terms, including privacy restrictions on real-robot recordings.

AI use statement

Generative AI tools assisted with manuscript drafting and editing, literature retrieval and reference checking, analysis and plotting code, methodological feedback, and results interpretation. The authors remain responsible for the accuracy of the manuscript and code.

References

Guillaume Alain and Yoshua Bengio. Understanding intermediate layers using linear classifier probes. In *International Conference on Learning Representations, Workshop Track*, 2017.

Nahid Alam. Causal physics steering in video world models via concept activation vectors. *arXiv preprint arXiv:2605.24322*, 2026.

Kushal Arora, Layla El Asri, Hareesh Bahuleyan, and Jackie Chi Kit Cheung. Why exposure bias matters: An imitation learning perspective of error accumulation in language generation. In *Findings of the Association for Computational Linguistics: ACL 2022*, pp. 700–710, 2022.

Mido Assran, Adrien Bardes, David Fan, Quentin Garrido, Russell Howes, Mojtaba Komeili, Matthew Muckley, Ammar Rizvi, Claire Roberts, Koustuv Sinha, Artem Zholus, et al. V-JEPA 2: Self-supervised video models enable understanding, prediction and planning. *arXiv preprint arXiv:2506.09985*, 2025.

Yonatan Belinkov. Probing classifiers: Promises, shortcomings, and advances. *Computational Linguistics*, 48(1):207–219, 2022.

Samy Bengio, Oriol Vinyals, Navdeep Jaitly, and Noam Shazeer. Scheduled sampling for sequence prediction with recurrent neural networks. In *Advances in Neural Information Processing Systems*, volume 28, 2015.

David Ha and Jürgen Schmidhuber. World models. *arXiv preprint arXiv:1803.10122*, 2018.

Danijar Hafner, Timothy Lillicrap, Ian Fischer, Ruben Villegas, David Ha, Honglak Lee, and James Davidson. Learning latent dynamics for planning from pixels. In *International Conference on Machine Learning*, pp. 2555–2565, 2019.

Danijar Hafner, Jurgis Pasukonis, Jimmy Ba, and Timothy Lillicrap. Mastering diverse domains through world models. *arXiv preprint arXiv:2301.04104*, 2023.

Nicklas Hansen, Hao Su, and Xiaolong Wang. TD-MPC2: Scalable, robust world models for continuous control. In *International Conference on Learning Representations*, 2024.

John Hewitt and Percy Liang. Designing and interpreting probes with control tasks. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pp. 2733–2743, 2019. doi: 10.18653/v1/D19-1275.

Siqiao Huang, Partha Kaushik, Michael Chen, Hengkai Pan, Kaiwen Geng, Omar Chehab, Fernando Moreno-Pino, and Max Simchowitz. Nano world models: A minimalist implementation of future video prediction. *arXiv preprint arXiv:2605.23993*, 2026.

Michael Janner, Justin Fu, Marvin Zhang, and Sergey Levine. When to trust your model: Model-based policy optimization. In *Advances in Neural Information Processing Systems*, volume 32, 2019.

Sonia Joseph, Quentin Garrido, Randall Balestriero, Matthew Kowal, Thomas Fel, Shahab Bakhtiari, Blake Richards, and Mike Rabbat. Interpreting physics in video world models. *arXiv preprint arXiv:2602.07050*, 2026.

Alexander Khazatsky, Karl Pertsch, Suraj Nair, Ashwin Balakrishna, Sudeep Dasari, Siddharth Karamcheti, Soroush Nasiriany, Mohan Kumar Srirama, Lawrence Yunliang Chen, Kirsty Ellis, et al. DROID: A large-scale in-the-wild robot manipulation dataset. *arXiv preprint arXiv:2403.12945*, 2024.

Alex M. Lamb, Anirudh Goyal, Ying Zhang, Saizheng Zhang, Aaron C. Courville, and Yoshua Bengio. Professor forcing: A new algorithm for training recurrent networks. In *Advances in Neural Information Processing Systems*, volume 29, 2016.

Hung Ming Liu. Probing the latent world: Emergent discrete symbols and physical structure in latent representations. *arXiv preprint arXiv:2603.20327*, 2026.

Lorenzo Mur-Labadia, Matthew Muckley, Amir Bar, Mido Assran, Koustuv Sinha, Mike Rabbat, Yann LeCun, Nicolas Ballas, and Adrien Bardes. V-JEPA 2.1: Unlocking dense features in video self-supervised learning. *arXiv preprint arXiv:2603.14482*, 2026.

Maxime Oquab, Timothée Darcet, Théo Moutakanni, Huy Vo, Marc Szafraniec, Vasil Khalidov, Pierre Fernandez, Daniel Haziza, Francisco Massa, Alaaeldin El-Nouby, et al. DINOv2: Learning robust visual features without supervision. *arXiv preprint arXiv:2304.07193*, 2023.

Samuele Punzo, Niccolò Caselli, Ippokratis Pantelidis, Francesco Massafra, Salvatore Lo Sardo, and Mohammadreza Salehi. How do video foundation models encode intuitive physics? Probing across pretraining paradigms. *arXiv preprint arXiv:2606.09646*, 2026.

Marc'Aurelio Ranzato, Sumit Chopra, Michael Auli, and Wojciech Zaremba. Sequence level training with recurrent neural networks. In *International Conference on Learning Representations*, 2016.

Stéphane Ross, Geoffrey Gordon, and Drew Bagnell. A reduction of imitation learning and structured prediction to no-regret online learning. In *International Conference on Artificial Intelligence and Statistics*, pp. 627–635, 2011.

Vlad Sobal, Wancong Zhang, Kyunghyun Cho, Randall Balestriero, Tim G. J. Rudner, and Yann LeCun. Learning from reward-free offline data: A case for planning with latent dynamics models. *arXiv preprint arXiv:2502.14819*, 2025.

Yu Sun, Xiaolong Wang, Zhuang Liu, John Miller, Alexei Efros, and Moritz Hardt. Test-time training with self-supervision for generalization under distribution shifts. In *International Conference on Machine Learning*, pp. 9229–9248, 2020.

Erik Talvitie. Model regularization for stable sample rollouts. In *Uncertainty in Artificial Intelligence*, pp. 780–789, 2014.

Basile Terver, Tsung-Yen Yang, Jean Ponce, Adrien Bardes, and Yann LeCun. What drives success in physical planning with joint-embedding predictive world models? *Transactions on Machine Learning Research*, 2026. URL `https://openreview.net/forum?id=cHZn5Gdh8e`.

Arun Venkatraman, Martial Hebert, and J. Andrew Bagnell. Improving multi-step prediction of learned time series models. In *Proceedings of the Twenty-Ninth AAAI Conference on Artificial Intelligence*, pp. 3024–3030, 2015. doi: 10.1609/aaai.v29i1.9590.

Dequan Wang, Evan Shelhamer, Shaoteng Liu, Bruno Olshausen, and Trevor Darrell. Tent: Fully test-time adaptation by entropy minimization. In *International Conference on Learning Representations*, 2021.

Ying Wang, Oumayma Bounou, Yann LeCun, and Mengye Ren. AdaJEPA: An adaptive latent world model. *arXiv preprint arXiv:2606.32026*, 2026.

Tianhe Yu, Deirdre Quillen, Zhanpeng He, Ryan Julian, Karol Hausman, Chelsea Finn, and Sergey Levine. Meta-World: A benchmark and evaluation for multi-task and meta reinforcement learning. In *Conference on Robot Learning*, pp. 1094–1100, 2020.

Wancong Zhang, Basile Terver, Artem Zholus, Soham Chitnis, Harsh Sutaria, Mido Assran, Randall Balestriero, Amir Bar, Adrien Bardes, Yann LeCun, and Nicolas Ballas. Hierarchical planning with latent world models. *arXiv preprint arXiv:2604.03208*, 2026.

Gaoyue Zhou, Hengkai Pan, Yann LeCun, and Lerrel Pinto. DINO-WM: World models on pre-trained visual features enable zero-shot planning. *arXiv preprint arXiv:2411.04983*, 2024.

A Additional results and design schematics

Reading the design schematics. These diagrams summarize the restart, adaptation, and input-perturbation designs. Quantitative panels in the main text use stored result arrays. The restart schematic retains the separate in-sample experiment's values; the main held-out restart comparison uses its own probe and sample split.

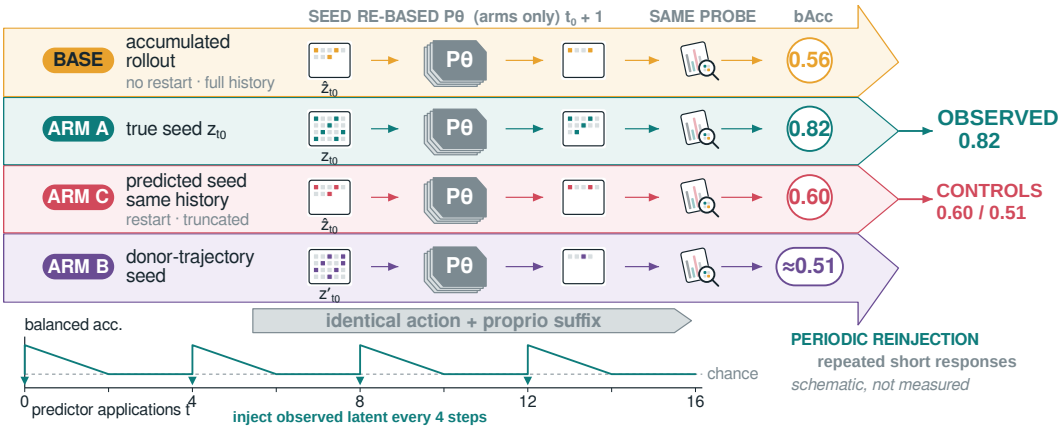


Figure 9: **Latent restart design and in-sample illustration.** Observed (A), predicted (C), and donor (B) restarts share truncated history; the uninterrupted path retains history. Rounded values 0.82/0.60/0.51/0.56 use a probe fitted and evaluated on all 240 trajectories. Their exact A–C difference is 0.226. Figure 6A gives the separate held-out comparison; its periodic panel uses the in-sample protocol illustrated here.

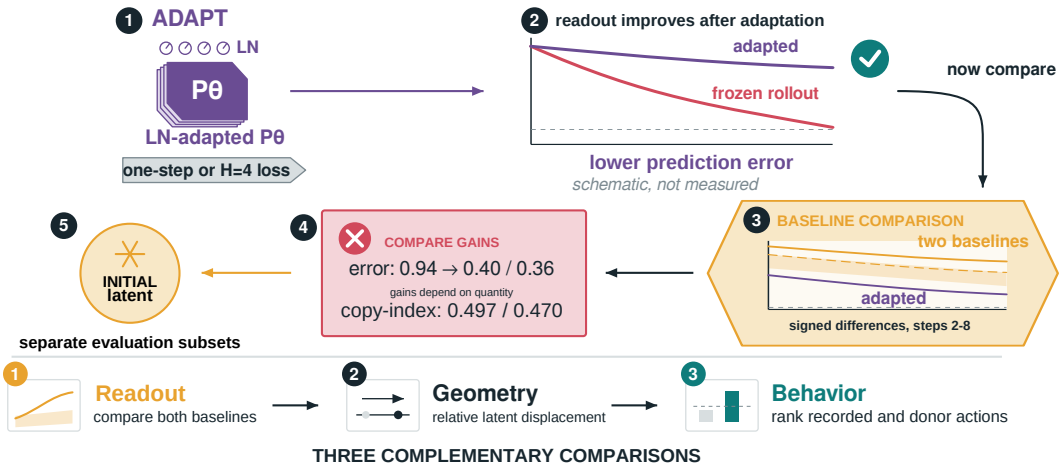


Figure 10: **Complementary adaptation comparisons.** Readout, geometry, and action ranking evaluate different properties. The drawn trajectories are schematic; Figure 8 separates measured end-effector and object readouts. Copy-index values 0.497 and 0.470 were evaluated on 10 and 20 trajectories, respectively, so they describe separate subsets. The main adaptation comparison uses matched evaluation sets and the same one-step error metric.

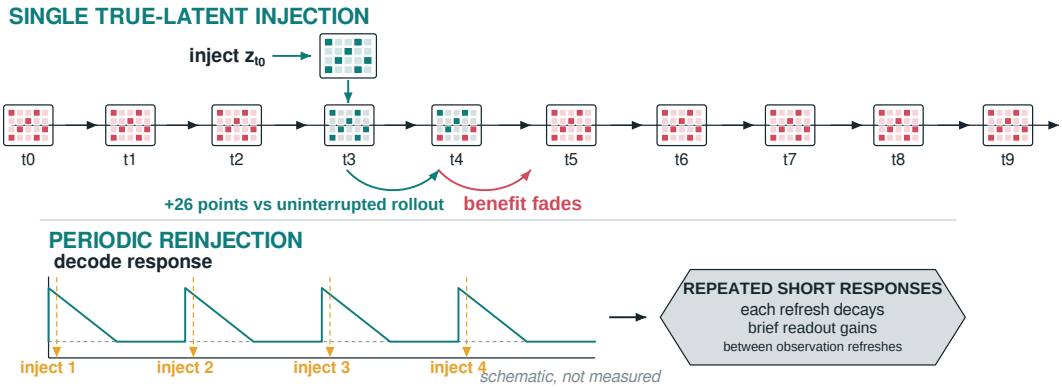


Figure 11: **Transient injection response.** The rounded 26-point label is the in-sample observed-restart minus uninterrupted contrast (0.823 versus 0.560). Its matched observed-minus-predicted contrast is 22.6 points. The sketch illustrates the response shape; the held-out matched contrast is independently 26.2 points in Figure 6A.

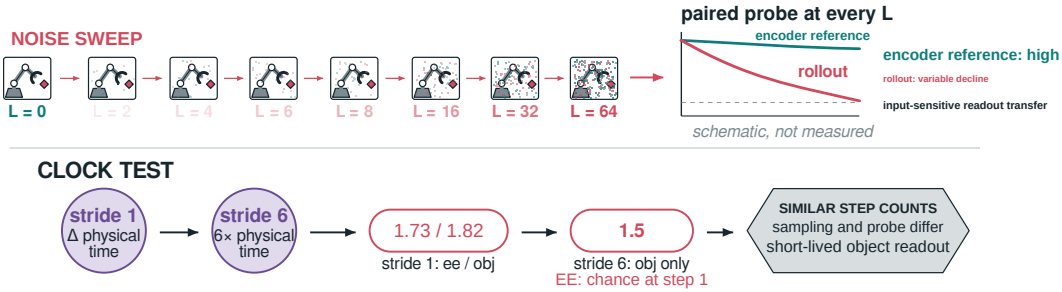


Figure 12: **Corruption and temporal sampling.** Noise perturbs a fixed native model with separately fitted probes. Stride changes physical time, conditioning, and selected PCA dimension. The diagram summarizes short object-readout half-lives of 1.82 and 1.50 predictions. End-effector decoding at stride 6 is already at chance at the first point. Measured noise responses are nonmonotonic (Figure 3C).

B PROTOCOL DETAILS

Rollout representation. The shifted-model protocol duplicates single frames to tubelet length and appends the final K predicted tokens at each step. In-distribution models use released interfaces and normalized native actions. Computation is float32 on a 24GB consumer GPU. A 16-frame window contains 15 prediction transitions; the 20-frame extension and physical-time mapping require the same distinction between frames and transitions.

Readout fitting and filtering. Grid pooling retains 4×4 spatial cells. The shifted encoder produces 22,528 pooled features before standardization and PCA. The initial dimension sweep compares 128, 256, and 512 components with a split drawn from the full trajectory set; this selection includes trajectories later used for evaluation. Final scaler, PCA, and classifier fits use training trajectories. Calibration and adaptation fix 128 components without that sweep. The native and stride surveys use the pooled observation-trained readout, whereas the exploratory periodic-restart probe uses all trajectories for fitting and scoring. Retained-class masks depend on evaluation labels at each horizon and are shared across paired paths.

Class baselines. Balanced accuracy averages recalls of retained labels. The shifted end-effector and object comparisons use a 0.5 reference. Native result metadata report four end-effector classes and three object classes, with nominal references 0.25 and $1/3$; gripper, contact, and task identity have nominal references $1/3$, 0.5, and 0.25. These native metadata count classes across the full window, while evaluation filtering is horizon-specific. Figure 3B therefore presents paired scores without imposing a shared chance line. Cross-model conclusions concern sustained within-model transfer, with class definitions kept distinct.

Half-life. For fixed chance c, the threshold is $[\mathrm{roll}(1) + c]/2$. Linear interpolation between consecutive measured scores locates the first crossing. Its coordinate $t_{1/2}$ is indexed from initialization at $t = 0$; elapsed time after the first prediction is $t_{1/2} - 1$. Full-precision shifted scores give 1.73 and 1.82 for end-effector and object position. Stride-6 object decoding gives 1.50; its end-effector score is 0.50 from the first prediction and supplies no positive initial gap for this definition.

Seed and bootstrap uncertainty. The noise statistic is mean balanced accuracy over predictions 1–15. At each positive noise level, error bars use the population standard deviation (divisor 3) of three perturbation-seed means on fixed trajectories. Level zero has one run. Bootstrap intervals use 1,000 trajectory resamples and the 2.5/97.5 percentiles, conditioning on the fitted probe. Horizon-wise intervals are pointwise. Action-ranking SE in the two-stride survey is population SD of binary wins divided by $\sqrt{240}$. Adaptation and restart curves are point estimates.

Restart protocols. The held-out restart analysis uses 120 training and 120 evaluation trajectories with a fixed PCA-128 observation readout. Its object A–C differences at steps 9–11 are 0.2623, 0.0952, and 0.0424. The in-sample analysis fits on all 240 trajectories and measures A/C/base/B as 0.8226/0.5968/0.5605/0.5139 at step nine. Its four-step refresh peaks at steps 5/9/13 are 0.8694/0.8226/0.8252. These protocols share the restart operator but differ in probe fitting; their curves remain separate. Calibration statistics fitted to the uninterrupted path characterize that path, and do not provide arm-specific calibration for the restarts.

Geometry statistic. Let Q be an orthonormal basis for the probe-weight subspace and define

$$s_Q(\hat{Z}) = \frac{\mathrm{mean}_d \,\mathrm{Var}[(\hat{Z}Q)_d]}{\mathrm{mean}_d \,\mathrm{Var}[(ZQ)_d]}. \tag{5}$$

The real contrast subtracts the mean ratio over 20 random subspaces of matching dimension. Each of 20 null draws forms $Z + \epsilon$, with isotropic errors scaled to each sample's rollout-error norm; its contrast uses five random subspaces. The reported standardized score is the real contrast minus the mean null contrast, divided by population SD of null contrasts. Positive values indicate a larger probe-versus-random variance contrast than this reference. The unequal 20/5 random-subspace averaging makes these descriptive standardized contrasts. They characterize spread across samples, while the probe accuracy assesses label recovery.

Calibration and continuous readouts. Diagonal and full affine maps use training observations and predictions at each horizon in an observation-fitted PCA-128 space. Ridge regularization is 1.0. The classifier is fitted once to pooled training observations. Continuous end-effector prediction uses ridge regression, evaluated by R^2 against both the initial true coordinate and a frozen-initial

readout. Its rollout-trained comparator is fitted using predicted training features. Paired intervals compare this readout with the stronger persistence baseline at each horizon.

Adaptation and ranking. One-step adaptation uses 120 trajectories times 15 transitions per phase. Four-step training uses 120 trajectories times four starting points. The 0.943-to-0.564 four-step-training comparison reports one-step evaluation error on the same 120 trajectories. Copy-index summaries use different subsets (10 before and 20 after) and are reported as separate descriptive values. For ranking, strict lower-cost wins are averaged; an exchangeable reference is $0.5(1 - p_{\mathrm{tie}})$. Donor actions are copied directly from another trajectory. The two-stride and held-out cost comparisons use different donor draws; within the held-out comparison the three costs use the same candidates.

C THE REAL-ROBOT EVALUATION

The DROID evaluation measures observation-reference readability on real-robot recordings. The reported data comprise 187 windows from 98 episodes. Encoder-reference end-effector decoding is 0.22–0.37 against four-class chance 0.25; gripper decoding is 0.42–0.54 against binary chance 0.5. Here, weak state decoding is already present in the encoded observations.

Camera placement varies across episodes, making the mapping between world-frame coordinates and image position scene-dependent. This is a candidate explanation for weak global position decoding. Per-episode probing is constrained by few windows, and gripper visibility may be limited by resolution. A scene-conditioned readout would test whether accounting for this variation improves the observation reference. DROID therefore identifies a readout-design boundary: the state must first be accessible from observations before temporal transfer can be assessed.

D NUMERICAL SUMMARIES

Table 1: **Worst-horizon adaptation differences, steps 2–8.** Each entry is the minimum of adapted balanced accuracy minus the specified baseline across seven horizons. Baselines are separate columns; quantities and adaptation protocols are separate rows. These are point estimates from stored curves.

Condition and quantity	Label persistence	Frozen latent
One-step inductive (EE)	-0.018	-0.020
One-step inductive (Object)	-0.028	-0.009
One-step transductive (EE)	-0.014	-0.009
One-step transductive (Object)	-0.028	-0.009
Four-step inductive (EE)	-0.046	-0.047
Four-step inductive (Object)	-0.017	$+0.002$

Table 2: **Held-out action ranking at horizon eight.** Costs are evaluated on identical recorded/donor candidate pairs for 120 trajectories. Brackets are pointwise 95% percentile bootstrap intervals.

Cost	Recorded-action win rate
Raw token ℓ_1	0.633 [0.550, 0.717]
PCA-space ℓ_2	0.542 [0.450, 0.633]
Calibrated PCA-space ℓ_2	0.392 [0.308, 0.483]

E RELATED EVALUATION QUESTIONS

Table 3: **Related evaluation questions.** These studies evaluate prediction, decoding, and planning. PRP adds paired observation and persistence comparisons, calibration tests, and matched restarts.

Work	Main question relevant to this study
HWM (Zhang et al., 2026)	Does hierarchical latent planning improve long-horizon control?
JEPA-WMs (Terver et al., 2026)	How do modeling choices affect planning success and rollout state-decoding error?
Nano World Models (Huang et al., 2026)	How do model and inference choices affect video prediction and rollout quality?
AdaJEPA (Wang et al., 2026)	Does adaptation within the control loop improve goal reaching?
Physics probing and steering (Alam, 2026; Joseph et al., 2026)	Where are physical variables readable, and how do interventions affect physical judgments?
This work	Which readouts transfer, how much does calibration recover, and what improves on persistence?