CONDITION-DEPENDENT PROBE CALIBRATION IN A WORLD MODEL'S OWN ROLLOUTS

Anonymous authors
Paper under double-blind review

ABSTRACT

Interpretability work on world models reads latents with probes calibrated on encoder outputs, then reuses them on latents the model generates with its own dynamics. Vakalis (2026) shows that read this way, the intervention effects a model imagines do not match the environment's. We ask whether that is lost information or a change in the measurement's own calibration, holding a TD-MPC2 model, its states and its edits fixed and varying only how the probe is calibrated. On perturbed cheetah states an encoder-calibrated readout falls from $R^2 = 0.910$ on encoded latents to 0.038 on latents rolled five steps to the same states, yet a map from rolled to encoded latents fit without the targets restores it to 0.823, short of the 0.846 reached by a regression refit on the initial latent and the actions. The map is specific to the condition it was fit in, keeping only part of its benefit at other horizons and doing worse than no map on another state distribution, and it is far from an isometry and 61% to 70% symmetric, so it is not a rotation of the latent frame. On cartpole, the one task whose sparse autoencoder passes the pipeline's own design checks, recalibration changes the Cohen's d separating features from controls by up to 2.6× across 4 comparisons and 1.35× in the pre-registered one, without reversing any feature-versus-control separation. In a second architecture, LeWorldModel on Reacher, the encoder-calibrated readout stays almost perfectly accurate on the model's training distribution, so the large TD-MPC2 mismatch is not universal across the settings tested.

1 INTRODUCTION

A learned world model compresses observations into a latent state and predicts how that state evolves under actions. Interpretability work reads those latents with probes, usually linear maps from the latent to a named quantity such as a joint angle or a velocity (Li et al., 2023; Nanda et al., 2023; Joseph et al., 2026; Zhang, 2026), and the readings are used to say what the model represents, what it predicts, and what an edit to its internals does.

The probe is almost always calibrated on latents the *encoder* produces from real observations, since those come with ground-truth state, and then applied to latents the model produces *recursively with its own dynamics* during an imagined rollout or a planning step. That reuse assumes the two populations express the physical variable in the same way.

Vakalis (2026) has already shown that the intervention effects a model imagines, read on rolled endpoints with a fixed encoder-calibrated probe, can be worse than predicting no effect at all, while the full predicted feature change stays aligned with the real one; that audit covers TD-MPC2, LeWorldModel and PreJEPA, and reads the result as distorted dynamics. We start from that failure rather than claim it, and treat calibration itself as the variable: with the model, states, actions and edits fixed, we ask whether the physical information is still recoverable, whether a target-agnostic correction restores compatibility, how it changes across rollout conditions, and whether recalibration alone changes a mechanistic measurement.

On perturbed cheetah states in TD-MPC2 (Hansen et al., 2024) an encoder-calibrated readout falls from $R^2 = 0.910$ on encoded latents to 0.038 on latents rolled five steps to the same states, yet a readout refit on the rolled latents reaches 0.839, and a linear map from rolled to encoded latents that never sees the targets brings the original readout back to 0.823 (Figure 1). Neither recovery readout beats a linear model that also sees the actions (0.846), which is fit fresh on inputs the readout never

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.

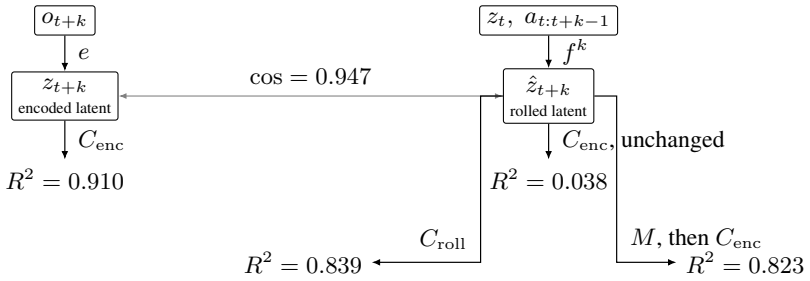

Figure 1: The measurement this paper makes, on perturbed cheetah states at $k = 5$ (means over three checkpoints, Table 3). Left: an observation is encoded and read by the probe C_{enc} calibrated on encoded latents. Right: the model rolls its own dynamics k steps to the same physical states and the same C_{enc} reads the result; the two latents stay close by cosine while the transferred readout loses most of its accuracy. Below: refitting on rolled latents (C_{roll}), or applying a target-agnostic realignment M before the *unchanged* C_{enc}, returns the measurement to 0.839 and 0.823 respectively, against the 0.846 of a regression refit on z_t and the actions. §5.3 shows M does not carry elsewhere.

sees; what recovery shows is that a fixed readout can be returned to just below it. The picture is task dependent, and on perturbed walker states the loss is not only calibration (§5.2).

The effect is not universal. In LeWorldModel (Maes et al., 2026) on Reacher, the encoder-calibrated readout transfers almost perfectly on the model's training distribution (§5.6), so what follows describes TD-MPC2 on three DeepMind Control tasks (Tassa et al., 2018), not learned rollouts in general.

Contributions. Vakalis (2026) establishes that imagined intervention effects come out distorted when read with a fixed encoder-calibrated probe. We treat that probe's calibration as part of the measurement model and vary it with the model and the interventions held fixed.

- **Calibration depends on the rollout condition.** A target-agnostic correction loses benefit with horizon distance and can do harm on another state distribution (§5.3).
- **The correction has a specific geometry,** matching what Dingeto (2026) report on a weight-update axis: far from an isometry, 61% to 70% symmetric, so not a rotation, with identity plus a rank-16 correction recovering 85% of its gain (§5.3).
- **Calibration moves a mechanistic measurement.** On cartpole, the one task whose sparse autoencoder passes the pipeline's own design checks, swapping the calibration source changes the Cohen's d separating intervention-consistency features from controls by up to 2.6× across 4 comparisons and 1.35× in the pre-registered one, and moves per-direction significance calls beyond refitting noise, where both readouts are accurate on the measured states and no feature-versus-control separation reverses (§5.5).
- **Scope and mechanism.** A whitened bound shows why a high latent cosine cannot certify a readout (§5.4); separately, both readouts stay above $R^2 = 0.973$ on the states IC reads in all 4 comparisons, and above 0.992 in the pre-registered one, while its effect sizes move, so pointwise accuracy does not certify a difference-based measurement either (§5.5); a second architecture where the readout transfers bounds the claim (§5.6).

2 Related work

Probing and its limits. Linear probes read information out of intermediate representations (Alain & Bengio, 2016), and probe accuracy alone is ambiguous: an expressive probe can learn the task rather than read it (Hewitt & Liang, 2019), decodable information need not be used by the model (Ravichander et al., 2021; Elazar et al., 2021), and results depend on the probe, the data and the baseline (Belinkov, 2022). Those critiques concern accuracy on the distribution the probe was fit on. We ask whether a probe calibrated on one *source* of representations, a world model's encoder, still measures the same quantity on another, the same model's dynamics. On other shift axes that

question already has a label-free affine repair, with the geometry we also find (Kirichenko et al., 2023; Csiszárik et al., 2021; Chen et al., 2025; Dingeto, 2026; Duan, 2026); Appendix J says what is and is not new here.

Probes inside world models. Probes have found emergent state representations in sequence models trained on games (Li et al., 2023; Nanda et al., 2023), physical quantities in video world models (Joseph et al., 2026), and latent structure in reinforcement-learning world models (Zhang, 2026); sparse autoencoders (Cunningham et al., 2024; Gao et al., 2024) supply candidate feature directions. Work reading a model's *predictions* through a probe fit on encoder outputs inherits the assumption we test. Singh (2026) reports the complementary case: latents a probe recovers almost perfectly that the planning objective cannot use.

Probes on rolled latents. That a fixed probe read on rolled latents can give a wrong answer is established; what the wrong answer implicates is not. Vakalis (2026) audits planning-time intervention fidelity across TD-MPC2 sizes, LeWorldModel and PreJEPA, with ridge probes fit on encoded latents and applied to rolled endpoints. On three LeWorldModel Cheetah checkpoints the predicted intervention effect is worse than predicting none: it keeps a contrast cosine of 0.85 to 0.89 with the real change while its projection on the task readout is nearly orthogonal and 1.7 to 2.6 times too large in magnitude, a ratio of imagined to real effect, and the pattern depends on model and task. Both the distortion and its model dependence are therefore prior work, on more models than we test. The questions differ: theirs is whether the imagined effect is faithful under the existing instrument, the probe never refit on rolled latents; ours is whether that instrument is still calibrated on the rolled representation, which lets us measure how the restoring correction depends on horizon and state distribution, what geometry it needs, and whether the calibration source moves a mechanistic metric.

Correcting or certifying physical quantities. Bao (2026) projects frozen DreamerV3 latents back onto a label-free invariant's level set, reducing rollout error, and Wang (2026) certifies how long a decoded invariant stays on its level set. Both repair or certify a physical quantity; we validate the calibration of a measurement function already fitted, and certify nothing.

Interventions measured through probes. Causal claims about internal directions are made by intervening and measuring downstream effects, through causal mediation (Vig et al., 2020), causal abstraction (Geiger et al., 2021), localised editing (Meng et al., 2022) and activation steering (Turner et al., 2023). Where that downstream effect is itself read by a probe on the intervened rollout, the probe's calibration is part of the reported effect size, which our IC experiments make explicit.

Rollout error and representation similarity. Latent world models (Ha & Schmidhuber, 2018; Hafner et al., 2023; Hansen et al., 2022; 2024) are trained to predict useful quantities, not to reproduce their encoder's outputs, and model-based reinforcement learning has long studied how multi-step error accumulates and relates to control performance (Janner et al., 2019; Lambert et al., 2020). A decoder fit on MuZero's encoded states likewise degrades on unrolled ones (Guei et al., 2024), and the gap between observation-anchored and free-running latents is usually attacked as a training objective (Shah & Shrestha, 2026; Li et al., 2026). That work measures error in the model's own units; we ask whether a fixed external measurement function survives the rollout, separating lost information from lost calibration. Representation similarity is usually summarised by a scalar: centred kernel alignment (Kornblith et al., 2019), which is dominated by high-variance structure, canonical correlation (Raghu et al., 2017), which is invariant to invertible linear maps, or latent cosine. None of the three reports the coordinate change a fixed probe feels, and a probe's sensitivity weights directions closer to the inverse of their variance (§5.4). TD-MPC2's simplicial normalisation (Lavoie et al., 2022) shapes the geometry where that difference lives.

3 Measuring probe transfer

3.1 Encoded and rolled latents

A TD-MPC2 world model has an encoder e mapping an observation o_t to a latent $z_t = e(o_t) \in \mathbb{R}^{512}$ and a latent dynamics model f with $z' = f(z, a)$. Both end in a simplicial normalisation that splits

the latent into 64 groups of 8 and applies a softmax within each group (Lavoie et al., 2022), so every entry is non-negative and each group sums to one. One model step spans two simulator steps (action repeat 2). For a state visited at time t with logged actions $a_t, \ldots, a_{t+k-1}$ we compare the *rolled* latent $\hat{z}_{t+k} = f(\cdots f(f(z_t, a_t), a_{t+1}) \cdots, a_{t+k-1})$ with the *encoded* latent $z_{t+k} = e(o_{t+k})$ of the observation actually reached. Every window lies inside one episode.

3.2 Readouts and diagnostics

For each task we define a readout target $q(x) \in \mathbb{R}^p$ from the raw simulator state x (Appendix B): positions and velocities of observable coordinates, with every unbounded hinge angle entered as its cosine and sine so that a rollout crossing $\pm\pi$ does not register a spurious jump. This gives $p = 5$ for cartpole, 18 for walker and 19 for cheetah. The **encoder-calibrated readout** C_{enc} is a ridge regression (10^{-4}, added to the Gram matrix) of $q(x_t)$ on z_t over training episodes. It is the estimator used by the IC analysis we build on (§3.5), and we use it unchanged.

A low R^2 for C_{enc} on rolled latents has two very different explanations: the rolled latent may no longer contain the physical state, or it may carry it in coordinates the readout does not match. Five further readouts separate them, all scored by R^2 against $q(x_{t+k})$ on held-out episodes, averaged over the p variables: C_{enc} applied to the encoded latent z_{t+k} (the accuracy the readout was fit for); a *rollout-calibrated* readout refit on $\hat{z}_{t+k}$; C_{enc} applied to $M\hat{z}_{t+k}$, where M is a *linear realignment*, a ridge map from rolled to encoded latents; and two baselines without the dynamics, a readout of $q(x_{t+k})$ from z_t alone and one from z_t with the k actions. The refit readouts and M are centred ridge regressions (10^{-2}) fit on training episodes separately at each horizon. We call the refit R^2 minus the encoder-calibrated R^2 on the same rolled latents the **transfer gap**. We repeat the encoder-calibrated and refit comparisons with a two-layer MLP to test whether a failure is specific to linear readouts.

3.3 Latent cosine against nulls

A natural latent-consistency check, and the one used to gate the IC analysis we build on, is the mean cosine between $\hat{z}_{t+k}$ and z_{t+k} against a threshold of 0.9. Simplicial latents are non-negative, so unrelated latents are not near cosine zero and the scale needs calibration. We report two nulls on the same windows: a *copy* null $\cos(z_t, z_{t+k})$, the score of a model that predicts no change, and a *shuffled* null pairing z_{t+k} with the encoded latent of an unrelated held-out window.

3.4 Where a readout can be hurt

Let the encoded training latents have sample covariance $\Sigma = V\Lambda V^\top$. We call the principal directions holding the last 10% of latent variance the *tail*, and for a rollout error $e = \hat{z}_{t+k} - z_{t+k}$ we report the share of $\|e\|^2$ in the tail beside the share of each readout's squared weight there. Cosine constrains the raw size of e; the next statement says which norm of e a readout actually feels.

For a centred ridge readout w of a scalar target q, fit on encoded latents with ridge ρ, Proposition 1 in Appendix A bounds the change the error causes: $|w^\top e| \le \sigma_q \min(\|\Lambda^{-1/2}V^\top e\|, \|e\|/2\sqrt{\rho})$, with σ_q the standard deviation of q. Readout error is controlled by a *whitened* norm of the latent error, which scales each direction by the inverse square root of its variance, while cosine sees only $\|e\|$ relative to $\|z\|$. A small raw error in low-variance directions can therefore leave cosine near one and still move the readout, and a larger ridge buys robustness at the cost of fit. The bound is a worst case over targets, so whether the error is realised depends on where the readout's weight lies, which we measure (§5.4).

3.5 Intervention consistency

Intervention consistency (IC) scores a direction u in a hooked activation of the dynamics model by editing the activation along u with magnitude δ, rolling the model forward H steps from a held-out context, and comparing the readout change $\Delta\hat{Q} = C(\hat{z}^{\text{edit}}_{t+1:t+H}) - C(\hat{z}_{t+1:t+H})$ with the change ΔQ a matched perturbation of size ϵ produces in the simulator, for the physical coordinate the direction is probed to represent. With the sign of ϵ matched per context, $\text{IC}(u) = \mathbb{E}[\langle \Delta\hat{Q}, \Delta Q\rangle / (\|\Delta\hat{Q}\|\,\|\Delta Q\| + \eta)]$ over contexts, with $\eta = 10^{-8}$ guarding against a zero norm. IC scores the *direction* of the predicted change and not its magnitude; the companion magnitude

metric is at its floor here for features and controls alike (Appendix F), so IC comparisons are comparisons of direction agreement only. The published comparisons use $\delta = 1.0$ at $H = 5$ and $\delta = 3.0$ at $H = 10$. Sparse-autoencoder feature directions are compared against three control sets of equal size: norm-matched random directions, random directions orthogonalised against the probe subspace, and directions matched to each feature's probe R^2. Separation is Cohen's d of features against each control set.

The readout C enters IC only through $C(\hat{z}^{\,\mathrm{edit}} - \hat{z})$, and only on rolled latents. We compare C_{enc} against a rollout-calibrated readout C_{roll}, a ridge regression (10^{-2}) on rolled latents of training episodes pooled over $k = 1, \ldots, 10$. The comparison is paired: identical features, contexts, control directions, sign matching and bootstrap seeds, differing only in the readout, with an assertion before analysis that every direction's identity, probed variable and probe R^2 agree across runs. The first comparison (checkpoint 1, $H = 5$), its outcomes and its predictions were pre-registered before any IC value under C_{roll} existed; the other three cartpole comparisons are replications, not confirmatory tests (Appendix C).

4 Experimental setup

Models and tasks. We use TD-MPC2 (Hansen et al., 2024) in its 5M-parameter single-task configuration on three DeepMind Control tasks (Tassa et al., 2018): cartpole-swingup, walker-walk and cheetah-run. For walker and cheetah we use the three released checkpoints per task; for cartpole, where no released checkpoint was used, two we trained (Table 1). Encoded and rolled latents are the 512-dimensional outputs of the encoder and of the dynamics model, and every rollout uses the model's own dynamics with the logged actions.

State distributions. Readout validity may depend on which states are visited, so every analysis runs on three distributions. *Planner*: the default initial states with actions from the model's own planner, as in evaluation. *Policy prior*: the default initial states with actions from the learned policy network alone, without planning; these states are reachable by the model, but its planner would not choose the actions. *Perturbed*: every unbounded hinge (the cartpole pole, the walker and cheetah torso) drawn uniformly over the full circle, other positions and velocities perturbed slightly, and actions from the policy prior with Gaussian noise ($\sigma = 0.3$), replaced by uniform random actions with probability 0.25. Episodes last 50 control steps.

Fitting, scoring and statistics. For every checkpoint and distribution all readouts and the realignment map are fit on 80% of episodes and scored on the remaining 20%, split by episode with a fixed seed, each fit and scored on the *same* distribution. A quantity reported for a task and distribution is the mean over the checkpoints in Table 1, with their range in brackets or as a band. IC uses a percentile bootstrap over contexts (10^4 resamples), and rank agreement across readouts is Spearman's ρ with a percentile bootstrap over directions. We report counts and ranges rather than significance tests, which two or three checkpoints would not support.

Intervention consistency. IC needs a sparse autoencoder on the hooked activation and per-feature probes. We have these for cartpole, where we reuse the SAE, probes, contexts and control directions of an existing IC analysis unmodified (two checkpoints, $H \in \{5, 10\}$, 50 top-probe features and 50 directions per control set); the readout swap in §5.5 changes nothing else. For cartpole the two checkpoints are two models we trained ourselves, stored as seeds in the released data and called checkpoints throughout. We ran the same pipeline on cheetah; §C says why it is uninformative.

5 Results

5.1 Encoder-calibrated readouts fail on rolled latents

We first measure how well an encoder-calibrated readout recovers absolute physical state from TD-MPC2's rolled latents, a different quantity from the imagined intervention effects audited by Vakalis (2026). On encoded latents it is accurate in every task and distribution (Figure 2, dotted grey); on rolled latents its accuracy depends on task, horizon and the states visited. On perturbed cheetah

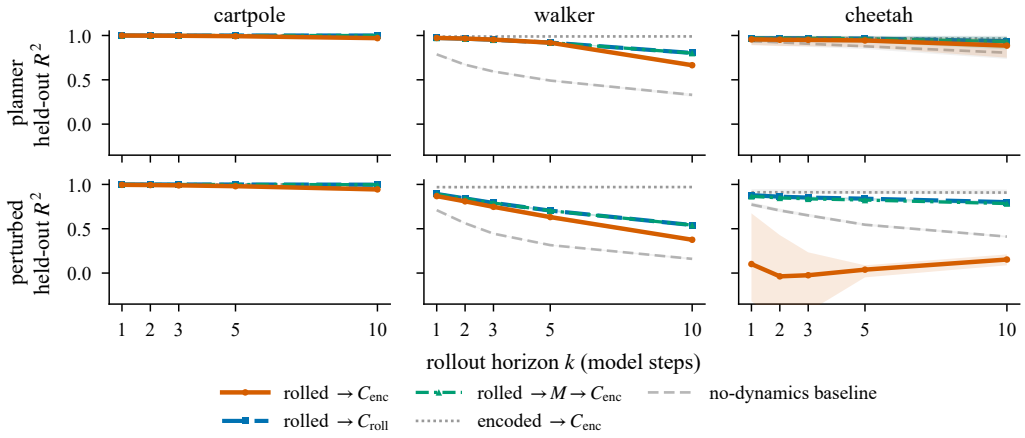

Figure 2: Mean held-out R^2 of physical readouts against rollout horizon k, by task (columns) for the planner and perturbed distributions (rows; the policy prior lies between them, Figure 4), averaged over checkpoints, with shading giving their range rather than a confidence interval. Full weight: the encoder-calibrated readout on rolled latents (rolled $\rightarrow C_{\text{enc}}$), a readout refit on rolled latents (C_{roll}), and the encoder-calibrated readout after the realignment M. Faint grey: the encoder-calibrated readout on encoded latents, and a no-dynamics baseline reading $q(x_{t+k})$ from z_t. Where the first curve falls away from the other two, the variables are present in the rolled latent but the readout cannot read them.

states at five model steps it reaches $R^2 = 0.038$ [−0.049, 0.088] across the three released checkpoints, against 0.910 on encoded latents of the same states, and after one step it varies widely across checkpoints (R^2 =-0.318 to 0.674).

We summarise the shortfall as the **transfer gap**: the R^2 of a readout refit on rolled latents minus that of the encoder-calibrated readout on the same latents. It depends on horizon, task, checkpoint and state distribution, and is not everywhere monotone. Under the planner and the policy prior it is largest at ten steps in both articulated bodies: 0.150 and 0.141 under the policy prior, 0.138 and 0.051 on the model's own planner states, against cartpole's 0.030 there. The effect is not confined to unusual states. On perturbed cheetah states it is large immediately (0.777 at one step) and peaks at $k = 2$, the worst horizon differing across checkpoints (Table 2). Cartpole is the negative control: its gap at ten steps is at most 0.081 on any checkpoint or distribution, against 0.648 on perturbed cheetah (Table 2), and distribution orderings are not universal across task, horizon and sampling (Appendix E). Both readouts see the same held-out states and differ only in calibration source, so this is not a train-test shift of the readout.

5.2 The measured variables remain recoverable on perturbed cheetah states

A readout can fail because the variable has left the latent or because the readout no longer matches the latent's coordinates. On perturbed cheetah states at $k = 5$ a readout refit on rolled latents reaches $R^2 = 0.839$ [0.832, 0.848], against 0.038 for the encoder-calibrated readout on the same latents, and a ridge map from rolled to encoded latents followed by the original unchanged readout reaches 0.823. That map never sees $q(x)$, so it tests whether a target-agnostic linear correction restores compatibility. The number of cheetah variables read at $R^2 \geq 0.9$ rises from 0 [0, 1] to 7 [6, 8] of 19 across the three checkpoints. The gap is not an artefact of regularisation, since an encoder-calibrated readout at the refit's ridge of 10^{-2} still scores 0.040, nor specific to linear probes: a two-layer MLP fit on encoded latents scores −0.818 on rolled latents against 0.892 for the same architecture refit on them.

The rolled latent carries a prediction, not a copy of the starting state: at ten steps the refit readout beats a readout of the same future state from z_t alone by 0.131 to 0.472 in R^2 across walker and cheetah distributions (Table 3). Neither recovery readout beats a linear model that also sees the actions: on perturbed cheetah at five steps that model reaches 0.846, against 0.839 for the refit and

0.823 for realignment. Its range across checkpoints (0.840 to 0.852) overlaps the refit's (0.832 to 0.848), and the ordering holds at one and ten steps. The comparison is not like for like, since that model is fit fresh on inputs the readout never sees while realignment leaves the readout unchanged, but it bounds what recovery shows: the fixed readout is returned to just below it, not past it. Recovery is substantial but not complete, and does not show the two representations are equivalent: realignment returns 0.823 of the 0.910 available on encoded latents. The picture is task dependent: on perturbed walker states at ten steps the encoder-calibrated readout falls from 0.970 to 0.376 and a refit on rolled latents reaches only 0.541 (Table 3), so there the loss is not only calibration. Held-out episode bootstrap intervals separate the encoder-calibrated readout from both recovery readouts on every checkpoint, and the two recovery intervals overlap, so we do not claim they differ (Table 9).

5.3 The correction is condition-specific and not a rotation

Within a condition the realignment works (§5.2); the question is whether it is one correction or many. We froze each fitted map, applied it at another horizon or state distribution of the same checkpoint with that condition's own readout, and scored the fraction of its gain the moved map recovers, suppressing the ratio where that gain is below 0.05 (Appendix E). On perturbed cheetah states recovery falls with distance along the horizon grid on every checkpoint, from 88% one step away to 35% four steps away, and no moved map in any condition recovers more than 96% (Figure 6). The decay tends to be asymmetric, a map fitted at the longer horizon transferring down better than the reverse in all 10 horizon pairs of the checkpoint mean but not on every checkpoint (Appendix E). Across state distributions, neighbouring ones exchange maps but distant ones do not, and the wrong map is worse than no map at all. On planner cheetah states, where the unmapped readout scores 0.938, prepending the map fitted on perturbed states drops it to -2.424; on perturbed states, where it scores 0.046, prepending the planner map drops it to -0.778. No single correction serves every condition, so the map diagnoses recoverability but cannot be fitted once and reused.

To see what the correction requires, we fitted fourteen constrained least-squares maps, in seven families each containing the identity, to the same target-agnostic objective (Appendix E; free parameters are the degrees of freedom of each affine map). The best orthogonal map recovers 40% (8% to 73%) of the full map's gain with 131,328 free parameters, against 53% (11% to 81%) for identity plus a rank-one correction with about 86 times fewer; the ranges overlap and the ordering reverses in three of the 5 conditions, so we do not claim the orthogonal family is worse, only that it is not better despite the parameter count. The negative conclusion rests instead on the map itself, which is far from an isometry (singular values from 17.4 down to numerically zero) and 61% to 70% symmetric, where a rotation near the identity would be dominated by its antisymmetric part (Appendix E). We therefore do not claim the change is a rotation, or the rolled latents a rotated frame. Identity plus a rank-16 correction recovers 85% of the gain on average (59% to 96% across conditions) with 6% of the full map's free parameters. The simplicial groups shape the correction without containing it, though we did not ablate it: within-group mass is 14 to 24 times chance, yet a free 8×8 map inside each group recovers 48%, against 69% for an identity-plus-rank-four correction with slightly fewer free parameters. These are five conditions at one horizon; we claim no characteristic rank and no law for how the correction changes.

5.4 High latent cosine does not certify a readout

Cosine stays high, and its scale is misleading. A latent-consistency check compares whole vectors while a probe reads one direction through them, and on our data the two disagree. On perturbed cheetah states at $k = 5$, rolled and encoded latents have mean cosine 0.947, above the 0.9 threshold, while the encoder-calibrated readout on those same latents scores 0.038; under the planner the pair is 0.995 and 0.944. A cosine check passes both, a readout separates them. The scale needs nulls too: simplicial latents are non-negative, so pairing each encoded latent with an unrelated one already gives 0.632 on cheetah, and a model predicting no change clears 0.9 on cartpole to five steps (0.952) and beats the dynamics model on one perturbed-cheetah checkpoint at $k = 1$ (Figure 5).

The disagreement is in low-variance directions. Encoded latents are far from isotropic: effective rank 13.3 of 512 on perturbed cheetah states, with 492 directions sharing the last 10% of variance, where the encoder-calibrated readout places 99% of its squared weight. Restricting the readout to the top-m principal directions localises the problem (Figure 8): at $k = 5$ on perturbed cheetah states

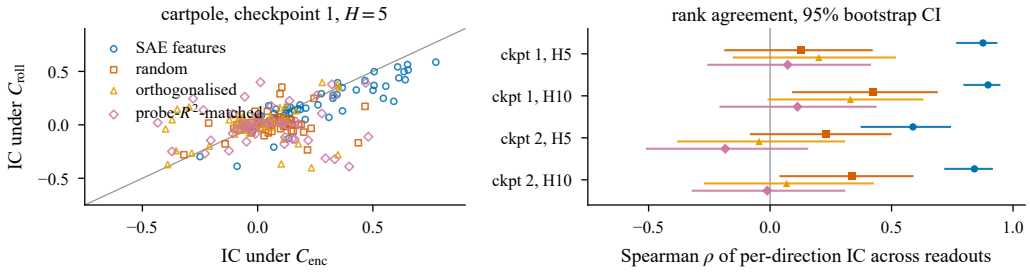

Figure 3: Intervention consistency (IC) on cartpole under two readouts. Left: IC of every direction for checkpoint 1 at $H = 5$, encoder-calibrated (x) against rollout-calibrated (y), with identical edits, contexts, perturbations and bootstrap seeds; colour and marker give the direction set, and the diagonal is equality. Right: Spearman rank correlation of per-direction IC between the readouts, with 95% bootstrap intervals. Features stay rank-consistent, control directions do not, and points move off the diagonal although the interventions are unchanged.

a readout on the top 8 directions loses only 0.118 in R^2 when moved to rolled latents, against 0.655 on the top 256, and 0.004 against 0.017 under the planner. The directions a readout needs for its last increments of accuracy are the ones the rollout disturbs, and ridge acts on the same trade-off, buying rolled accuracy at the cost of encoded accuracy from 10^{-4} to 10^{3} (Figure 8).

Why an aggregate metric cannot certify a probe. Proposition 1 bounds the change a rollout error e can cause in a ridge readout by $\sigma_q \|\Lambda^{-1/2} V^\top e\|$, the error in whitened coordinates, where a direction counts in inverse proportion to the square root of its variance; cosine constrains $\|e\|$ relative to $\|z\|$ and is blind to that rescaling. The bound held on every sample we checked (Appendix A); as a worst case over targets it explains why a high cosine cannot certify a readout, not how much a given readout loses. Tail concentration is not sufficient either: it holds under every distribution, including those where readouts transfer well (Appendix E). What separates the failures is the size of that error and the readout's dependence on those directions. Cosine still says *which* conditions are worst while certifying none: its rank correlation with the gap over 130 conditions is -0.808, holds within every horizon and on a held-out task, yet its magnitude error exceeds the gap's own spread (Appendix K). Without targets one can triage, not certify. For probes of language models, whitened similarity likewise predicts probe transfer where raw cosine does not (Ying et al., 2026a;b); Proposition 1 instantiates the point for ridge readouts under a model's own rollout error.

5.5 Readout calibration changes measured intervention effects on cartpole

The readout enters IC only through $C(\hat{z}^{\,\mathrm{edit}}) - C(\hat{z})$ (§3.5): for a linear readout, its weights on a small difference between two rolled latents. We claim nothing about whether a probe direction is causal; the claim concerns the instrument that turns an intervention into a number. Cartpole has the smallest transfer gap of our three tasks, so it is where IC should be least sensitive to the readout. We pre-registered, for checkpoint 1 at $H = 5$, the prediction that IC would be nearly unchanged (Appendix C), then repeated it at $H = 10$ and on a second checkpoint.

Both readouts are accurate where IC operates, and the measurement still moves. On the exact 50 contexts of the pre-registered comparison, rolled five steps, the encoder-calibrated readout reaches mean $R^2 = 0.992$ and the rollout-calibrated one 0.998 (Appendix F). Swapping them still moves feature-versus-orthogonalised separation from $d = +1.405$ to $+1.044$, and control directions lose most of their rank agreement (Figure 3). Across all 4 comparisons the magnitudes move in no consistent direction (Table 8). The largest change is 2.6×, on checkpoint 2 at $H = 5$, a replication whose gate dropped exactly the two variables on which its encoder-calibrated readout fails it, leaving three of five for both readouts, so the restriction is forced by the pipeline rather than chosen (Appendix C). In the pre-registered comparison the largest change is 1.35×, and both readouts pass a post hoc validity check on that comparison's states (Appendix F). A paired bootstrap over directions puts the ratio's 95% interval above 1 in 7 of the 12 contrasts (4 comparisons by three control sets), so the change is not direction-sampling noise, though the intervals are too wide to

size it. Six of the seven are the checkpoint 2 contrasts, which read three variables rather than five; the one like-for-like contrast that survives is the pre-registered feature-versus-orthogonalised one, at 1.35× (Table 6). No feature-versus-control separation changed sign, and the companion magnitude metric is at its floor for features and controls alike under both readouts (Appendix F), so every IC comparison here is one of direction agreement only. Per-direction conclusions do change: whether a direction's IC interval lies above, below or across zero differs between readouts for 9 of 50 features and 26 to 31 of 50 controls per set, well beyond refitting one estimator on disjoint halves (a post hoc count; Appendix F). Feature IC stayed rank-consistent across readouts (Spearman ρ 0.588 to 0.897); control IC did not (ρ −0.184 to 0.423, most intervals covering zero), and mean feature IC fell under the rollout-calibrated readout in all four comparisons. The pattern matches the mechanism of §5.4 at intervention scale: the edits displace the rolled latent by a median of at most 2.81% of its norm but up to 14.8 in whitened units, and two readouts that agree on absolute state need not agree on displacements of that kind.

A separately pre-registered test, written down after the first comparison, separates the calibration source from refitting noise: we fit both readouts on each of two disjoint halves of the training episodes (checkpoint 1, $H = 5$) and compared IC within and across sources. Refitting inside one source left per-direction IC close to unchanged (feature $\rho \geq 0.960$), while swapping source on the same half gave feature $\rho \leq 0.905$ and larger changes (Appendix F gives the control correlations and the change magnitudes). An exploratory sweep, planned after the main result, finds the change at every edit magnitude over an eightfold range with no sign change (Appendix F).

5.6 The readout transfers in a second architecture, LeWorldModel on Reacher

We ran the core diagnostic unchanged on the official pretrained LeWorldModel for Reacher (Maes et al., 2026), on its training distribution (Appendix I). The encoder-calibrated readout transfers: from one to ten model steps it scores $R^2 \geq 0.994$ on rolled latents against ≥ 0.998 for the rollout-calibrated readout, a gap of at most 0.004, while the no-dynamics baseline falls to 0.495. This is the regime where TD-MPC2 also looks fine, so one task, one checkpoint and no perturbed states cannot separate architecture from task or state distribution; what it shows is that the large TD-MPC2 mismatch is not universal across the settings tested.

6 Discussion

Vakalis's audit shows imagined intervention effects coming out distorted under a fixed readout, on more models than we test. It motivates a measurement question it does not resolve: whether the instrument is still calibrated on the rolled representation. Read that way the failure is a property of the pairing between probe and rollout condition rather than of the probe, and it is more specific than generic degradation: a map that never sees the targets could not restore destroyed information, yet within a condition it restores most of the readout, while across conditions it weakens with horizon, can harm another state distribution, and is not a rotation (§5.3). Capacity does not explain the recovery either, since identity plus a rank-16 correction recovers 85% of the full map's gain with 6% of its parameters while the best orthogonal map recovers 40%. The contribution is not that a fixed probe can fail on rollouts, but that probe calibration is condition-dependent and can move a mechanistic measurement while, on the task where we measure it most closely, the physical information stays recoverable.

No feature-versus-control separation changes sign in any comparison or swept magnitude, but effect sizes, control-direction rank agreement and per-direction significance calls do change with the calibration source, beyond refitting noise and where both readouts are accurate on the states measured (§5.5). The practical step is cheap: report the transfer gap, or without targets rank conditions by latent cosine (Appendix K), and repeat intervention measurements under a second calibration.

Limitations. Results are TD-MPC2 at one size, three tasks, two or three checkpoints each, so we report ranges and bootstrap intervals, not significance tests. LeWorldModel on Reacher shows no mismatch, but on its training distribution alone, so it cannot separate architecture from task, distribution or simplicial normalisation. Clean IC evidence is cartpole only, walker and cheetah each failing a gate, so no cell shows both the collapse and its mechanistic effect (Appendix F).

REPRODUCIBILITY STATEMENT

Every number, table and figure in this paper is regenerated from the stored rollouts and checkpoints by `scripts/run_extension_pipeline.sh`, and every measured value in the main text is a generated macro rather than a transcribed one; a small number of per-condition values quoted inline in the appendices are transcribed from the same released files (Appendix L). The readout targets are defined in Appendix B, the fitting, scoring and splitting protocol in §4, and the pre-registration, its deviations and the outcomes against prediction in Appendix C. Proposition 1 is proved in Appendix A and its inequality is checked numerically on every held-out sample.

REFERENCES

Guillaume Alain and Yoshua Bengio. Understanding intermediate layers using linear classifier probes. *arXiv preprint arXiv:1610.01644*, 2016.

Richard Bao. Correcting a Learned Physical Invariant Improves World-Model Rollouts. *arXiv preprint arXiv:2608.23526*, 2026.

Yonatan Belinkov. Probing Classifiers: Promises, Shortcomings, and Advances. *Computational Linguistics*, 2022. arXiv:2102.12452.

Alan Chen, Jack Merullo, Alessandro Stolfo, and Ellie Pavlick. Transferring Linear Features Across Language Models With Model Stitching. *arXiv preprint arXiv:2506.06609*, 2025.

Adrián Csiszárik, Péter Kőrösi-Szabó, Ákos K. Matszangosz, Gergely Papp, and Dániel Varga. Similarity and Matching of Neural Network Representations. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2021. arXiv:2110.14633.

Hoagy Cunningham, Aidan Ewart, Logan Riggs, Robert Huben, and Lee Sharkey. Sparse Autoencoders Find Highly Interpretable Features in Language Models. In *International Conference on Learning Representations (ICLR)*, 2024. arXiv:2309.08600.

Hiskias Dingeto. Train the Model, Not the Reader: Decodability Supervision for Verifiable Activation Explanations. *arXiv preprint arXiv:2607.20379*, 2026.

Evan Duan. Do Activation Monitors Survive Model Updates? Benchmarking, Predicting, and Repairing Activation-Monitor Staleness. *arXiv preprint arXiv:2606.15980*, 2026.

Yanai Elazar, Shauli Ravfogel, Alon Jacovi, and Yoav Goldberg. Amnesic Probing: Behavioral Explanation with Amnesic Counterfactuals. *Transactions of the Association for Computational Linguistics*, 2021. arXiv:2006.00995.

Leo Gao, Tom Dupré la Tour, Henk Tillman, Gabriel Goh, Rajan Troll, Alec Radford, Ilya Sutskever, Jan Leike, and Jeffrey Wu. Scaling and evaluating sparse autoencoders. *arXiv preprint arXiv:2406.04093*, 2024.

Atticus Geiger, Hanson Lu, Thomas Icard, and Christopher Potts. Causal Abstractions of Neural Networks. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2021. arXiv:2106.02997.

Hung Guei, Yan-Ru Ju, Wei-Yu Chen, and Ti-Rong Wu. Demystifying MuZero Planning: Interpreting the Learned Model. *IEEE Transactions on Artificial Intelligence*, 2024. arXiv:2411.04580.

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

Danijar Hafner, Jurgis Pasukonis, Jimmy Ba, and Timothy Lillicrap. Mastering Diverse Domains through World Models. *arXiv preprint arXiv:2301.04104*, 2023.

Nicklas Hansen, Xiaolong Wang, and Hao Su. Temporal Difference Learning for Model Predictive Control. *arXiv preprint arXiv:2203.04955*, 2022.

Nicklas Hansen, Hao Su, and Xiaolong Wang. TD-MPC2: Scalable, Robust World Models for Continuous Control. In *International Conference on Learning Representations (ICLR)*, 2024. arXiv:2310.16828.

John Hewitt and Percy Liang. Designing and Interpreting Probes with Control Tasks. In *Empirical Methods in Natural Language Processing (EMNLP)*, 2019. arXiv:1909.03368.

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 (NeurIPS)*, 2019. arXiv:1906.08253.

Sonia Joseph, Quentin Garrido, Randall Balestriero, Matthew Kowal, Thomas Fel, Shahab Bakhtiari, Blake Richards, and Mike Rabbat. Interpreting Physics in Video World Models. In *International Conference on Machine Learning (ICML)*, 2026. arXiv:2602.07050.

Polina Kirichenko, Pavel Izmailov, and Andrew Gordon Wilson. Last Layer Re-Training is Sufficient for Robustness to Spurious Correlations. In *International Conference on Learning Representations (ICLR)*, 2023. arXiv:2204.02937.

Simon Kornblith, Mohammad Norouzi, Honglak Lee, and Geoffrey Hinton. Similarity of Neural Network Representations Revisited. In *International Conference on Machine Learning (ICML)*, 2019. arXiv:1905.00414.

Nathan Lambert, Brandon Amos, Omry Yadan, and Roberto Calandra. Objective Mismatch in Model-based Reinforcement Learning. *arXiv preprint arXiv:2002.04523*, 2020.

Samuel Lavoie, Christos Tsirigotis, Max Schwarzer, Ankit Vani, Michael Noukhovitch, Kenji Kawaguchi, and Aaron Courville. Simplicial Embeddings in Self-Supervised Learning and Downstream Classification. *arXiv preprint arXiv:2204.00616*, 2022.

Jiaqi Li, Xinglong Zhang, Haibin Xie, Yixing Lan, Wei Pan, and Xin Xu. Koopman Dreamer: Spectrally Constrained Latent Dynamics for Stable World-Model Imagination. *arXiv preprint arXiv:2607.19719*, 2026.

Kenneth Li, Aspen K. Hopkins, David Bau, Fernanda Viégas, Hanspeter Pfister, and Martin Wattenberg. Emergent World Representations: Exploring a Sequence Model Trained on a Synthetic Task. In *International Conference on Learning Representations (ICLR)*, 2023. arXiv:2210.13382.

Lucas Maes, Quentin Le Lidec, Damien Scieur, Yann LeCun, and Randall Balestriero. LeWorld-Model: Stable End-to-End Joint-Embedding Predictive Architecture from Pixels. *arXiv preprint arXiv:2603.19312*, 2026.

Kevin Meng, David Bau, Alex Andonian, and Yonatan Belinkov. Locating and Editing Factual Associations in GPT. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2022. arXiv:2202.05262.

Neel Nanda, Andrew Lee, and Martin Wattenberg. Emergent Linear Representations in World Models of Self-Supervised Sequence Models. *arXiv preprint arXiv:2309.00941*, 2023.

Maithra Raghu, Justin Gilmer, Jason Yosinski, and Jascha Sohl-Dickstein. SVCCA: Singular Vector Canonical Correlation Analysis for Deep Learning Dynamics and Interpretability. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2017. arXiv:1706.05806.

Abhilasha Ravichander, Yonatan Belinkov, and Eduard Hovy. Probing the Probing Paradigm: Does Probing Accuracy Entail Task Relevance? In *European Chapter of the Association for Computational Linguistics (EACL)*, 2021. arXiv:2005.00719.

Rishi Shah and Rishav Shrestha. Rollout-Decoded Reconstruction for Long-Horizon Prediction in Latent World Models. *arXiv preprint arXiv:2608.25017*, 2026.

Joyjeet Singh. The Objective Is the Bottleneck: Latent World Models Encode What Their Planners Cannot Use. *arXiv preprint arXiv:2608.12959*, 2026.

Yuval Tassa, Yotam Doron, Alistair Muldal, Tom Erez, Yazhe Li, Diego de Las Casas, David Budden, Abbas Abdolmaleki, Josh Merel, Andrew Lefrancq, Timothy Lillicrap, and Martin Riedmiller. DeepMind Control Suite. *arXiv preprint arXiv:1801.00690*, 2018.

Alexander Matt Turner, Lisa Thiergart, Gavin Leech, David Udell, Juan J. Vazquez, Ulisse Mini, and Monte MacDiarmid. Steering Language Models With Activation Engineering. *arXiv preprint arXiv:2308.10248*, 2023.

Donna Vakalis. The Intervention Gap in Latent World Models. *arXiv preprint arXiv:2608.29998*, 2026.

Jesse Vig, Sebastian Gehrmann, Yonatan Belinkov, Sharon Qian, Daniel Nevo, Simas Sakenis, Jason Huang, Yaron Singer, and Stuart Shieber. Causal Mediation Analysis for Interpreting Neural NLP: The Case of Gender Bias. *arXiv preprint arXiv:2004.12265*, 2020.

Hongbo Wang. When Do Conservation Laws Survive Learned Representations? Certified Horizons for Latent World Models. *arXiv preprint arXiv:2606.24945*, 2026.

Zhuofan Josh Ying, Peter Hase, and Nikolaus Kriegeskorte. Comparing Linear Probes with Mahalanobis Cosine Similarity. *arXiv preprint arXiv:2606.19603*, 2026a.

Zhuofan Josh Ying, Shauli Ravfogel, Nikolaus Kriegeskorte, and Peter Hase. The Truthfulness Spectrum Hypothesis. *arXiv preprint arXiv:2602.20273*, 2026b.

Xinyu Zhang. What Do World Models Learn in RL? Probing Latent Representations in Learned Environment Simulators. *arXiv preprint arXiv:2603.21546*, 2026.

DISCLOSURE OF LLM USAGE

Large language models (Claude, Anthropic) were used in preparing this paper, in the categories declared on the submission form: to aid and polish writing; for retrieval and discovery, including searching for related work and checking bibliography entries against their primary sources; for research ideation and execution, including proposing analyses and writing the scripts that produce them; to draft sections of the paper; and in work on the proof of Proposition 1. They were not used to generate synthetic data.

What that does not extend to. Every quantitative claim is produced by the released pipeline rather than by a model: `scripts/run_extension_pipeline.sh` regenerates every number, table and figure from the stored rollouts and checkpoints, and every measured value in the main text is a generated macro (Appendix L), so no reported value is model-written. The experiments, the preregistration in Appendix C and its recorded deviations, and the decision of what to claim are the authors'. Proposition 1 was checked by hand and its inequality verified numerically on every held-out sample (Appendix A). Every bibliography entry was verified against live primary sources, with the retrieved pages archived in the supplementary bundle. The authors reviewed all model-assisted text, code and analysis, and take full responsibility for the content of the paper.

A THE WHITENED ERROR BOUND

Proposition 1 (whitened error bound)**.** *Let w be the centred ridge readout of a scalar target q on encoded latents, $w = V(\Lambda + \rho I)^{-1}V^\top s$ with s the sample covariance of z and q and $\rho \geq 0$. For any latent perturbation e,*

$$\left|w^\top e\right| \;\leq\; \sigma_q \left\|\Lambda^{1/2}(\Lambda+\rho I)^{-1}V^\top e\right\| \;\leq\; \sigma_q \min\Big(\left\|\Lambda^{-1/2}V^\top e\right\|,\; \tfrac{\|e\|}{2\sqrt{\rho}}\Big),$$

where σ_q is the sample standard deviation of q and $\Lambda^{-1/2}$ is taken on the support of Λ.

Let $Z \in \mathbb{R}^{n\times d}$ hold centred encoded latents and $q \in \mathbb{R}^n$ a centred scalar target. Write $\Sigma = Z^\top Z/n = V\Lambda V^\top$, $s = Z^\top q/n$ and $\sigma_q^2 = q^\top q/n$. The centred ridge readout is $w = (\Sigma+\rho I)^{-1}s = V(\Lambda+\rho I)^{-1}V^\top s$; when $\rho = 0$ the inverse is taken on the support of Λ.

Step 1. s lies in the column space of $Z^\top$, which is the range of Σ, so $s = \Sigma^{1/2}a$ with $a = (\Sigma^{1/2})^+ s$. Then $\|a\|^2 = s^\top\Sigma^+ s = q^\top Z(Z^\top Z)^+ Z^\top q/n = \|Pq\|^2/n \leq \sigma_q^2$, where P is the orthogonal projection onto the column space of Z.

Table 1: Checkpoints, readout dimension p, and the number of checkpoints analysed on each state distribution. Each episode has 50 control steps; 20% of episodes are held out.

				checkpoints analysed		
task	source	available	p	planner	policy prior	perturbed
cartpole-swingup	trained	2	5	2	2	2
walker-walk	released	3	18	3	3	3
cheetah-run	released	3	19	3	3	3
episodes per checkpoint				200	1000	1000

Step 2. For any $e \in \mathbb{R}^d$,

$$w^\top e = a^\top \Sigma^{1/2} V(\Lambda + \rho I)^{-1} V^\top e = (V^\top a)^\top \Lambda^{1/2}(\Lambda + \rho I)^{-1} V^\top e,$$

using $\Sigma^{1/2} = V\Lambda^{1/2}V^\top$. By Cauchy–Schwarz and orthogonality of V, $|w^\top e| \le \|a\| \, \|\Lambda^{1/2}(\Lambda + \rho I)^{-1}V^\top e\| \le \sigma_q \, \|\Lambda^{1/2}(\Lambda + \rho I)^{-1}V^\top e\|$.

Step 3. The matrix $\Lambda^{1/2}(\Lambda + \rho I)^{-1}$ is diagonal with entries $c_i = \lambda_i^{1/2}/(\lambda_i + \rho)$, and $c_i = 0$ when $\lambda_i = 0$. For $\lambda_i > 0$, $\lambda_i + \rho \ge \lambda_i$ gives $c_i \le \lambda_i^{-1/2}$, and for $\rho > 0$ the inequality $\lambda_i + \rho \ge 2\sqrt{\lambda_i \rho}$ gives $c_i \le 1/(2\sqrt{\rho})$. Hence $\|\Lambda^{1/2}(\Lambda + \rho I)^{-1}V^\top e\|$ is at most $\|\Lambda^{-1/2}V^\top e\|$, with $\Lambda^{-1/2}$ on the support, and at most $\|V^\top e\|/(2\sqrt{\rho}) = \|e\|/(2\sqrt{\rho})$. □

Numerical check. `experiments/fidelity/verify_bound.py` fits the centred-ridge analogue of the encoder-calibrated readout (ridge 10^{-4} on the Gram matrix, i.e. $\rho = 10^{-4}/n$) on a random 80% of episodes and evaluates the first inequality for every variable on $e = \hat{z}_{t+5} - z_{t+5}$ over 1800 to 6000 held-out windows per setting, for 7 checkpoint and distribution pairs. It held on every sample in every setting (minimum fraction 1.000), and $\|a\| \le \sigma_q$ held for every variable. On the seed 1 checkpoints used here, the median readout change $|w^\top e|/\sigma_q$ was ordered like the transfer gaps within each task, and the median whitened norm $\|\Lambda^{-1/2}V^\top e\|$ was largest on perturbed states but did not separate the planner from the policy prior on cheetah. For cheetah the two values, whitened norm then readout change, are 12.5 and 0.047 under the planner, 12.3 and 0.127 under the policy prior, 19.1 and 0.309 on perturbed states; for walker 7.1 and 0.073 (planner), 15.4 and 0.173 (policy prior), 21.0 and 0.274 (perturbed). The median latent cosine was above 0.9 in every setting. Perturbed cartpole had a whitened norm of 151.0 and a readout change of only 0.039. Cartpole has the largest whitened error and the smallest readout change: the bound limits what a readout *can* suffer, and cartpole's readouts do not load on the directions its rollouts err in. The bound is slack where it holds: the per-setting median ratio of the realised change to the bound runs 3.2×10^{-4} to 0.016, so it supports the qualitative point that cosine is blind to the whitening rescaling and not any quantitative account of how much a readout loses.

B Readout targets

Targets are computed from MuJoCo's raw state $[q_{\text{pos}}, q_{\text{vel}}]$ by `src/envspec.py`. **cartpole** ($p = 5$): cart position, cosine and sine of pole angle, cart velocity, pole angular velocity. **walker** ($p = 18$): torso height, cosine and sine of torso pitch, six joint angles (hip, knee, ankle, each leg), height velocity, horizontal velocity, torso angular velocity, six joint velocities. Absolute horizontal position is excluded because the observation does not contain it. **cheetah** ($p = 19$): torso height, cosine and sine of torso pitch, five joint angles, cosine and sine of the back foot angle (its joint range extends past $-\pi$), height velocity, horizontal velocity, torso angular velocity, six joint velocities. Absolute horizontal position is excluded for the same reason.

C Pre-registration and deviations

The IC readout-calibration comparison was written down and committed before any IC value under C_{roll} existed (`experiments/ic_frame/PREREGISTRATION.md`). It specified: C_{enc} as the

existing estimator and, for cartpole, the existing readout file; C_{roll} as a centred ridge (10^{-2}) on rolled latents pooled over $k = 1, \ldots, 10$; the same rolled-R^2 gate threshold (0.9, worst case over $k \in \{1, 3, 5, 10\}$); one cartpole run at seed 1 and $H = 5$ with everything else as in the existing analysis; a cheetah run as a diagnostic only, with all 19 variables kept under both readouts since C_{enc} passes none on that distribution; Spearman correlation, Cohen's d and mean absolute change as outcomes; and the predictions that cartpole IC would be nearly unchanged and cheetah's would change substantially.

Deviations. (i) After the seed 1, $H = 5$ result we added three further cartpole comparisons (checkpoint 1 at $H = 10$; checkpoint 2 at $H = 5$ and $H = 10$) using the same analysis, at the same edit magnitude within a horizon but not across them: the published comparisons use $\delta = 1.0$ at $H = 5$ and $\delta = 3.0$ at $H = 10$. They were not pre-registered and are replications, not confirmatory tests. Because H and δ covary across the four comparisons, we claim no horizon trend from Table 8. (ii) The checkpoint 2 IC analysis being replicated kept only three variables under its gate (cosine and sine of the pole angle, pole angular velocity), so both checkpoint 2 runs use those three variables for both readouts; without this the directions would not be paired, which our pairing assertion detected before any comparison was computed. The gate drops exactly the two variables on which checkpoint 2's encoder-calibrated readout fails it (cart position 0.872, cart velocity 0.829), so the restriction is forced by the pipeline rather than chosen, and it applies to the comparison carrying the largest reported change. (iii) The cheetah sparse autoencoder failed one of the pipeline's own design checks (maximum feature firing rate 0.633, threshold 0.5). We ran the pre-registered diagnostic anyway, as specified, and report it in Appendix F as a diagnostic only. (iv) A within-calibration null was pre-registered separately after the first comparison (Appendix F).

Outcome against prediction. Both predictions were wrong. Cartpole: the sign of separation was preserved in all comparisons, but IC was not "nearly unchanged" (Table 8). The prediction failed in magnitude and for controls. Cheetah: the prediction of large per-direction changes also failed, for a reason the prediction did not anticipate. That sparse autoencoder failed one of the pipeline's own design checks (maximum feature firing rate 0.633 against a threshold of 0.5) and its features were indistinguishable from the control sets under *both* readouts (Cohen's d from -0.026 to $+0.270$), so there was little signal for the readout to move and the small change between readouts is not evidence that readout choice is harmless there (Appendix F). We make no claim about cheetah's features. Appendix G lists interpretations of our own that these measurements overturned.

D FULL FIDELITY TABLES

Table 2 gives the transfer gap for every task, distribution and horizon in the main text. Tables 3 and 4 give every quantity behind Figures 2–5, as means over the n checkpoints analysed. Per-checkpoint values are in `data_runs/results/fidelity_summary.csv`. MLP readouts were computed at $k \in \{1, 5\}$ only.

Table 2: Transfer gap: the R^2 of a readout refit on rolled latents minus the R^2 of the encoder-calibrated readout on the same rolled latents, at horizon k. Mean over checkpoints, with the range in brackets where more than one checkpoint was analysed. Larger means more of the readout's accuracy is lost to calibration rather than to missing information.

task	distribution	$k = 1$	$k = 5$	$k = 10$
cartpole	planner	0.001 [0.000, 0.001]	0.008 [0.006, 0.009]	0.030 [0.025, 0.035]
cartpole	policy prior	0.001 [0.001, 0.001]	0.009 [0.005, 0.012]	0.039 [0.026, 0.052]
cartpole	perturbed	0.004 [0.002, 0.007]	0.019 [0.009, 0.029]	0.053 [0.026, 0.081]
walker	planner	0.000 [−0.001, 0.000]	0.001 [−0.005, 0.004]	0.138 [0.128, 0.154]
walker	policy prior	0.018 [0.014, 0.023]	0.056 [0.048, 0.065]	0.150 [0.141, 0.162]
walker	perturbed	0.025 [0.018, 0.033]	0.073 [0.060, 0.088]	0.166 [0.157, 0.177]
cheetah	planner	0.013 [0.001, 0.034]	0.021 [0.003, 0.058]	0.051 [0.021, 0.111]
cheetah	policy prior	0.009 [0.006, 0.015]	0.038 [0.025, 0.063]	0.141 [0.081, 0.223]
cheetah	perturbed	0.777 [0.192, 1.182]	0.801 [0.745, 0.885]	0.648 [0.598, 0.693]

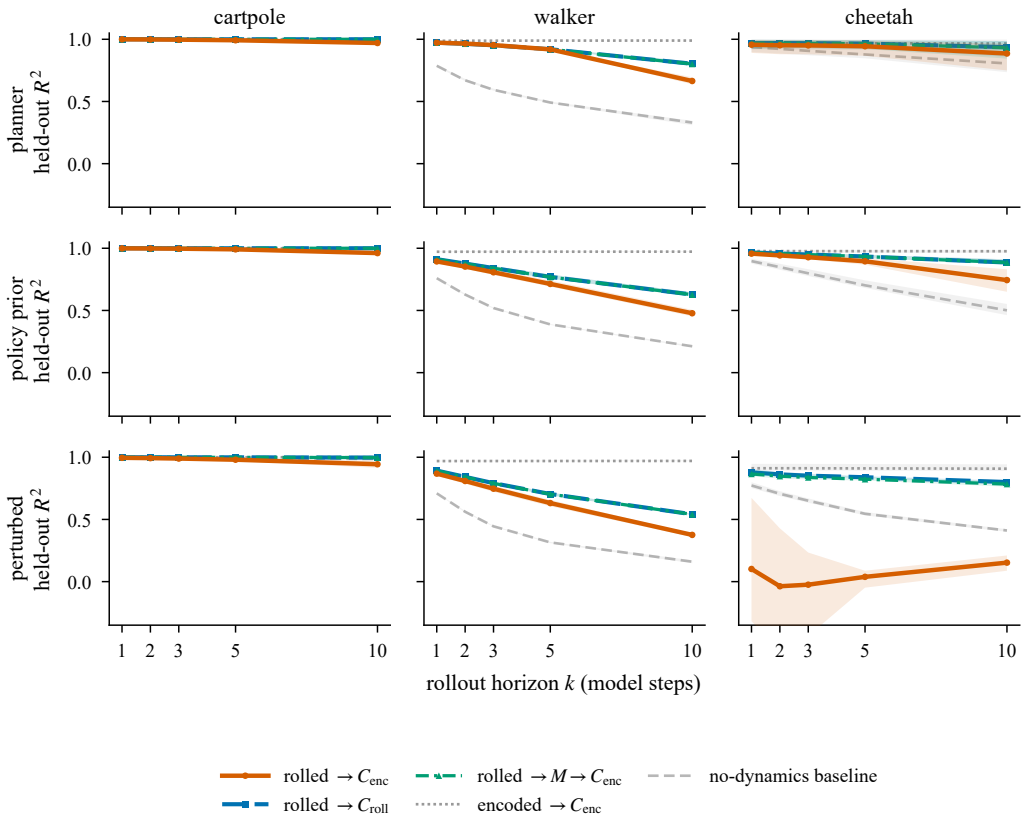

Figure 4: Figure 2 with the policy-prior distribution included, all three distributions as rows. Lines as in Figure 2.

E GEOMETRY AND IDENTIFIABILITY

Transfer of the realignment map. §5.3 freezes each fitted map and applies it elsewhere. The full grid is 1350 (source, target) pairs: two tasks, three checkpoints, three state distributions and five horizons, crossed, in `data_runs/results/m_transfer_long.csv`, with the same-distribution and same-horizon slices split out into `m_cross_horizon.csv` and `m_cross_distribution.csv`. Each file carries R^2 for the transferred map, for no map, for the target's own map, for the rollout-calibrated readout and for the encoded-latent ceiling, so the suppressed recovery fractions can be checked directly. As a correctness check, all 90 same-condition cells reproduce the published encoder-calibrated, rollout-calibrated and realigned values, to a maximum absolute difference of zero. On perturbed cheetah states mean recovery is 88%, 72%, 53% and 35% at distances one to four along the horizon grid, and it decreases with distance on each checkpoint separately. For each of the 10 pairs of horizons we compared the map fitted at the longer horizon applied at the shorter with the reverse: the longer-horizon map transfers better in all 10 in the checkpoint mean, and in 10, 6 and 8 on checkpoints 1, 2 and 3, so the asymmetry is a tendency rather than a rule. On walker the map fitted on policy-prior states restores perturbed-state R^2 to 0.742 against 0.686 unmapped, and the reverse direction reaches 0.795 against 0.749, so adjacent distributions do exchange maps.

Why we do not transfer maps between checkpoints. The checkpoints per task are independently trained, so nothing ties dimension i of one to dimension i of another. We measured this rather than assuming it (`experiments/fidelity/ckpt_comparability.py`). A readout fitted on one checkpoint's encoded latents, the setting it was built for, scores 0.895 or better within checkpoint but at most -15.271 on another cheetah checkpoint's encoded latents, and at most -1.038 on walker.

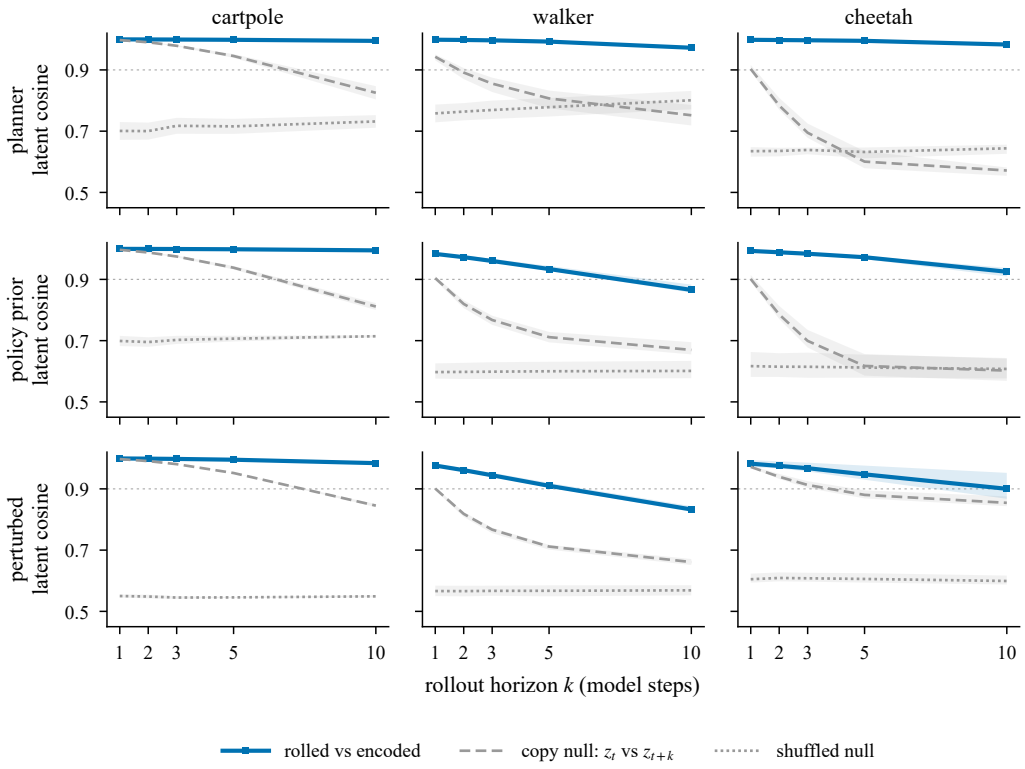

Figure 5: Mean cosine between rolled and encoded latents (blue), against a copy null that predicts no change (dark grey) and a shuffled null (light grey). The thin line marks the 0.9 threshold of the IC analysis we build on. Rows are state distributions; means over checkpoints, bands show the range.

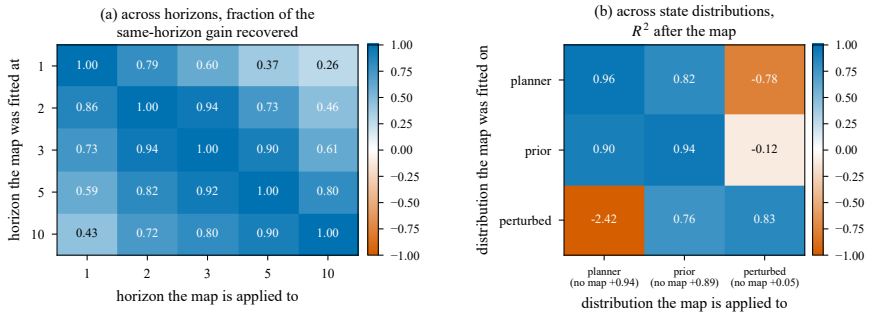

Figure 6: A fitted realignment map does not carry over. **(a)** Cheetah, perturbed states, mean of three checkpoints: the fraction of the same-horizon gain a map recovers when frozen and applied at another horizon. **(b)** Raw R^2 when a map is moved between state distributions, with the unmapped value under each column; the scale is clipped at ± 1. Both panels use the target condition's own unchanged readout.

Per-dimension latent means correlate across checkpoints at $|r| \leq 0.041$ on cheetah and 0.075 on walker. A cross-checkpoint map would therefore be fitting an alignment between two arbitrary bases, and its score would measure that alignment rather than any transfer, so we do not report one.

Which map families suffice. Table 5 and Figure 7 compare the fourteen fits at $H = 5$, each the least-squares solution of the same target-agnostic objective, rolled to encoded latents on training

Table 3: Mean held-out R^2 of readouts. enc. on enc.: encoder-calibrated readout on encoded latents; enc. on rolled: on rolled latents; refit: ridge refit on rolled latents; realigned: encoder-calibrated readout after a ridge map from rolled to encoded latents; z_t only and z_t + actions: baselines without the dynamics model.

task	dist.	k	n	enc. on enc.	enc. on rolled	refit	realigned	z_t only	z_t + actions	MLP enc.	MLP refit
cartpole	planner	1	2	1.000	0.999	1.000	1.000	1.000	1.000	0.999	1.000
		5	2	1.000	0.992	1.000	1.000	1.000	1.000	0.994	1.000
		10	2	1.000	0.970	1.000	1.000	1.000	1.000	–	–
cartpole	prior	1	2	1.000	0.999	1.000	1.000	0.999	1.000	0.999	1.000
		5	2	1.000	0.991	1.000	1.000	0.999	0.999	0.994	1.000
		10	2	1.000	0.961	1.000	1.000	0.998	0.999	–	–
cartpole	perturbed	1	2	1.000	0.995	1.000	1.000	0.998	0.999	0.999	1.000
		5	2	1.000	0.980	0.999	0.999	0.992	0.997	0.983	0.999
		10	2	1.000	0.944	0.997	0.997	0.986	0.995	–	–
walker	planner	1	3	0.988	0.973	0.972	0.972	0.787	0.852	0.975	0.975
		5	3	0.989	0.919	0.920	0.920	0.491	0.643	0.923	0.927
		10	3	0.990	0.665	0.803	0.801	0.331	0.534	–	–
walker	prior	1	3	0.972	0.894	0.912	0.911	0.759	0.835	0.916	0.942
		5	3	0.972	0.713	0.770	0.768	0.388	0.606	0.731	0.798
		10	3	0.973	0.478	0.627	0.627	0.212	0.510	–	–
walker	perturbed	1	3	0.969	0.868	0.893	0.891	0.710	0.817	0.891	0.925
		5	3	0.970	0.632	0.704	0.703	0.315	0.568	0.649	0.729
		10	3	0.970	0.376	0.541	0.541	0.160	0.470	–	–
cheetah	planner	1	3	0.969	0.956	0.969	0.968	0.938	0.962	0.980	0.986
		5	3	0.967	0.944	0.965	0.964	0.878	0.938	0.970	0.979
		10	3	0.966	0.886	0.937	0.931	0.806	0.903	–	–
cheetah	prior	1	3	0.977	0.957	0.966	0.965	0.897	0.955	0.979	0.988
		5	3	0.977	0.895	0.933	0.933	0.702	0.881	0.916	0.963
		10	3	0.976	0.744	0.885	0.885	0.501	0.816	–	–
cheetah	perturbed	1	3	0.911	0.102	0.878	0.866	0.773	0.895	−0.480	0.939
		5	3	0.910	0.038	0.839	0.823	0.545	0.846	−0.818	0.892
		10	3	0.908	0.153	0.801	0.784	0.411	0.807	–	–

episodes, under a different constraint. Every family contains the identity. **Parameter counting.** Every map is affine, $\hat{z} \mapsto (\hat{z} - \bar{\hat{z}})W + \bar{z}$, and we count its free parameters as the degrees of freedom of the constrained family W belongs to, plus $d = 512$ for the offset: $d(d-1)/2$ for an orthogonal W, $2dr - r^2$ for a rank-r correction to the identity (the dimension of the rank-r matrices, not the $2dr$ entries of its stored factors), d for a diagonal W, $64 \cdot 8^2$ for the simplicial block-diagonal family and d^2 for the dense ridge map. The same definition is used in Table 5, Figure 7 and §5.3. Level 1 metrics (latent cosine and relative latent error) and Level 2 (the unchanged readout's R^2) move together but not proportionally: on perturbed cheetah states at $H = 5$ the orthogonal map lifts checkpoint 1's latent cosine from 0.930 to 0.968, still short of the 0.976 that checkpoint 2 reaches with no map at all, while checkpoint 1's readout after that map (0.141) sits above checkpoint 2's no-map readout (0.088): a higher Level 1 score does not buy a higher Level 2 score, which is another instance of §5.4's point. One caveat matters for reading the negative result about rotations: because every family minimises the same latent objective, "the best orthogonal map" means the best in latent reconstruction, not the rotation that would maximise the readout's R^2. We did not search for the latter, since a map chosen using the targets would no longer be target-agnostic and could not be compared with the others. Task dependence is real and we do not smooth it over: on perturbed walker states, where the whole gain is only 0.07 in R^2, the block family beats the rank-one correction, and the ordering of the cheap families flips.

Structure of the full map. Across the 5 conditions its largest singular value runs to 17.4 and its smallest is numerically zero; effective rank by singular-value entropy is 299 to 336 of 512, with the top eight directions holding 25% to 49% of its energy and the top 64 holding 66% to 80%. Its relative distance from the identity is 0.9 to 1.5, so it is not a small perturbation, and 61% to 70% of its energy is symmetric. The dense map is therefore far from low rank. The low-rank result in §5.3 says that a moderate-rank correction to the identity recovers most of what the *readout* needs, not that the two representations differ along a few directions in general, and not that any low-dimensional mechanism underlies the difference. Mass within the simplicial groups is 22% to 38%, which is 14 to 24 times the chance share of $8/512$, leaving 62% to 78% crossing groups. Full values are in `data_runs/results/m_structure.json`.

Table 4: Latent agreement and geometry. cos: mean cosine of rolled and encoded latents; copy and shuf.: nulls (§3.3); rel. err.: $\|\hat{z}_{t+k} - z_{t+k}\|/\|z_{t+k}\|$, and the same for the copy prediction z_t; err. in tail: share of squared rollout error in the directions holding the last 10% of encoded variance; loss m: R^2 lost from encoded to rolled latents by a readout on the top-m principal directions.

task	dist.	k	n	cos	cos copy	cos shuf.	rel. err.	rel. err. copy	err. in tail	loss m=8	loss m=256
cartpole	planner	1	2	1.000	0.997	0.701	0.021	0.069	0.908	0.000	0.001
		5	2	0.998	0.945	0.716	0.054	0.322	0.906	0.006	0.008
		10	2	0.995	0.825	0.732	0.094	0.577	0.908	0.023	0.030
cartpole	prior	1	2	1.000	0.997	0.699	0.022	0.077	0.847	0.001	0.001
		5	2	0.998	0.938	0.707	0.054	0.338	0.809	0.005	0.009
		10	2	0.995	0.811	0.714	0.095	0.599	0.797	0.025	0.039
cartpole	perturbed	1	2	0.999	0.997	0.550	0.032	0.069	0.747	0.000	0.002
		5	2	0.995	0.952	0.546	0.084	0.288	0.537	0.005	0.018
		10	2	0.984	0.845	0.549	0.151	0.524	0.473	0.023	0.053
walker	planner	1	3	0.999	0.943	0.758	0.042	0.314	0.547	−0.003	0.014
		5	3	0.992	0.807	0.778	0.095	0.601	0.406	−0.008	0.066
		10	3	0.972	0.752	0.801	0.191	0.696	0.329	0.082	0.319
walker	prior	1	3	0.983	0.904	0.597	0.145	0.407	0.398	0.004	0.075
		5	3	0.934	0.711	0.600	0.305	0.736	0.268	0.024	0.254
		10	3	0.866	0.670	0.601	0.467	0.797	0.217	0.094	0.488
walker	perturbed	1	3	0.976	0.901	0.566	0.187	0.420	0.365	0.009	0.098
		5	3	0.910	0.712	0.567	0.383	0.739	0.239	0.045	0.332
		10	3	0.833	0.661	0.569	0.546	0.808	0.195	0.117	0.586
cheetah	planner	1	3	0.998	0.903	0.635	0.036	0.410	0.527	0.002	0.006
		5	3	0.995	0.601	0.632	0.069	0.886	0.512	0.004	0.017
		10	3	0.983	0.572	0.644	0.129	0.926	0.511	0.018	0.071
cheetah	prior	1	3	0.993	0.901	0.617	0.088	0.420	0.418	0.005	0.019
		5	3	0.972	0.617	0.612	0.189	0.863	0.315	0.033	0.080
		10	3	0.925	0.602	0.608	0.326	0.883	0.261	0.116	0.230
cheetah	perturbed	1	3	0.982	0.972	0.605	0.143	0.172	0.508	0.026	0.471
		5	3	0.947	0.880	0.606	0.265	0.382	0.297	0.118	0.655
		10	3	0.900	0.854	0.599	0.375	0.435	0.233	0.215	0.608

Table 5: Map families at $H = 5$, as the mean fraction of the full ridge map's gain recovered over 5 conditions, with the range across them. Free parameters are degrees of freedom of each affine map, including its 512-dimensional offset (see text). Source: `data_runs/results/m_geometry.csv`.

map	free parameters	mean	range
no map	0	0%	
mean shift	512	18%	[2%, 35%]
diagonal affine	1,024	35%	[15%, 59%]
identity + rank 1	1,535	53%	[11%, 81%]
identity + rank 2	2,556	60%	[14%, 86%]
SimNorm blocks ($64 \times 8{\times}8$)	4,608	48%	[28%, 67%]
identity + rank 4	4,592	69%	[23%, 91%]
identity + rank 8	8,640	77%	[39%, 94%]
identity + rank 16	16,640	85%	[59%, 96%]
identity + rank 32	32,256	91%	[75%, 97%]
identity + rank 64	61,952	95%	[88%, 98%]
identity + rank 128	115,200	98%	[94%, 99%]
orthogonal (Procrustes)	131,328	40%	[8%, 73%]
full ridge	262,656	100%	

Tail concentration is common. The share of squared rollout error lying in the lowest-variance directions that hold 10% of encoded variance exceeds 10% in every task, distribution and horizon (Table 4), including the planner distribution, where readouts transfer. Likewise the encoder-calibrated readout's squared weight in those directions is 94% on cartpole under the planner, 81% on walker and 97% on cheetah. Concentration of error and of readout weight in the tail is therefore not by itself a sign of failure. What differs on perturbed cheetah states is the size of the error: relative rollout error at $k = 1$ is 0.143, against 0.036 under the planner, approaching the error of the copy prediction (0.172), which it exceeds on one of the three checkpoints.

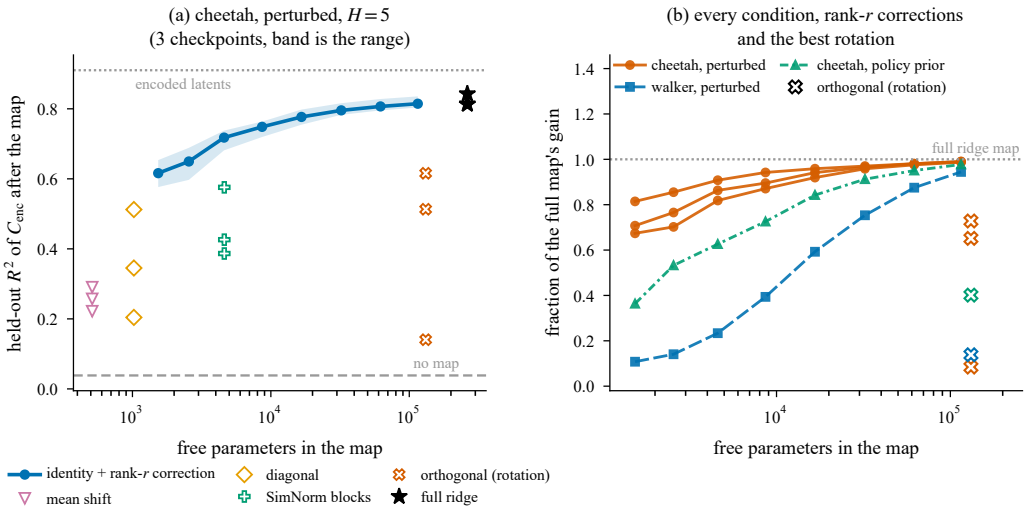

Figure 7: How much structure the correction needs, at $H = 5$. **(a)** Cheetah, perturbed states: held-out R^2 of the unchanged encoder-calibrated readout after each map, against the map's free parameters, with the three checkpoints' range shaded. **(b)** Every condition, as a fraction of the full ridge map's gain. The best orthogonal map (crosses) sits at or below the rank-r curve despite about 86 times more free parameters, though the ordering reverses in three of the 5 conditions, so we claim only that it is not better (§5.3).

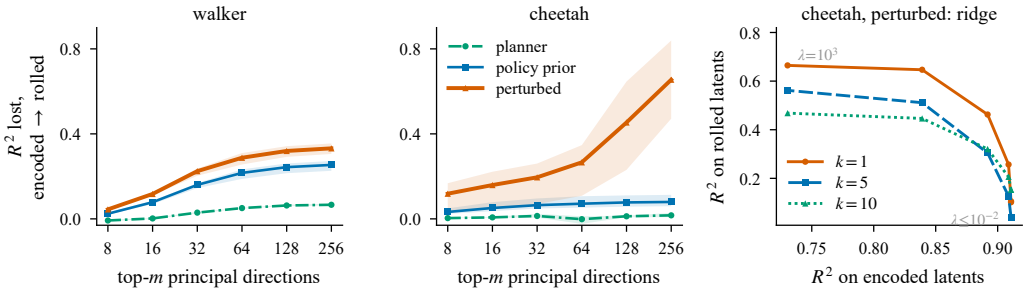

Figure 8: Where the two representations disagree, at $k = 5$. Left and centre: readouts restricted to the top-m principal directions of encoded latents, and the R^2 they lose when moved from encoded to rolled latents, per state distribution (colours). High-variance directions transfer; the loss appears as lower-variance directions enter, and stays small under the planner. Right: on perturbed cheetah states, raising the ridge from 10^{-4} to 10^3 costs encoded accuracy and buys rolled accuracy, one line per horizon. Means over checkpoints, bands the range.

Distribution dependence, and its limits. Which states a rollout visits changes the size of the gap, but not by a law we can state cleanly. Within a checkpoint and horizon, walker's gap is smallest under the planner and largest on perturbed states in all 15 of 15 pairs; on cheetah, perturbed states beat the policy prior in all 15 pairs, but the planner's gap falls below the policy prior's in only 11, and cartpole shows no consistent ordering. Sample size matters too, and matching it reverses walker's planner-versus-prior ordering at ten steps (Appendix E). Distribution shift can amplify the mismatch, and interventional analyses visit unusual states, but the ordering is not universal across task, horizon and sampling.

What else differs between the distributions. The effective rank of encoded cheetah latents is 17.9, 31.1 and 13.3 under the planner, policy prior and perturbed states, which does not follow the ordering of the gaps, so diversity measured this way does not explain them. Sample size explains

part of one ordering. With the policy prior cut to the planner's 200 episodes (seed 1), cheetah's gap at ten steps is unchanged (0.211, against 0.223 with all episodes and 0.022 under the planner), while walker's shrinks: it stays above the planner's to five steps (0.029 against 0.004) and falls below it at ten (0.102 against 0.128).

Identifiability. Readout weights in low-variance directions are poorly determined by the data. Refitting the existing cartpole seed 1 readout with the same data, split and ridge in double precision changes individual weights by up to 2.9×10^{-3}, almost entirely (100%) within the 502 tail directions, while the two readouts' predictions on encoded test latents differ by only 6.1×10^{-8} target standard deviations. On latents rolled ten steps they differ 19.6 times as much (1.2×10^{-6}). The magnitudes are negligible here, but the mechanism is the one in §5.4: agreement on encoded latents does not constrain the tail, and rolled latents expose it.

F INTERVENTION CONSISTENCY: DETAILS

The two readouts used for IC were gated on held-out cartpole episodes. On seed 1 the encoder-calibrated readout's mean rolled R^2 was 0.998, 0.990 and 0.971 at $k = 1, 5, 10$, and the rollout-calibrated readout's 0.998, 0.999 and 0.996; on seed 2, 0.993, 0.969, 0.918 and 0.996, 0.997, 0.989. Both readouts are accurate; they differ most at the longest horizon, where the IC edits are read.

Both readouts are accurate where IC actually operates. Those gates are measured on all held-out windows. IC does not use all of them: it samples 50 held-out (episode, timestep) pairs from 46 episodes and rolls H steps with the recorded actions. `experiments/ic_frame/ic_state_validity.py` rebuilds exactly those contexts, with the same data file, split, horizon, context count and seed, and scores both readouts on the latents IC reads. On those latents the encoder-calibrated readout reaches mean $R^2 = 0.992$ (worst variable 0.984) and the rollout-calibrated readout 0.998 (worst 0.994), a transfer gap of 0.006. Cartpole's readout is therefore close to valid in the exact neighbourhood the interventions probe, and the effect-size changes in §5.5 are not produced by a broken instrument in an extreme region. They arise because IC reads a *difference* between two nearly identical rolled latents, so a readout discrepancy that is negligible for absolute R^2 need not be negligible for $\Delta\hat{q}$.

The other three comparisons (post hoc). We applied the same scoring, unchanged, to the remaining comparisons (`experiments/ic_frame/ic_state_validity_all.py`). The IC runs did not store context identities, so the contexts are the deterministically reconstructed ones: the IC script's own sampler called with the same data, split, horizon, count and seed, consistent with each run's logged context count. Mean R^2 for the encoder-calibrated and rollout-calibrated readouts is 0.973 and 0.996 at checkpoint 1, $H = 10$; 0.992 and 0.997 at checkpoint 2, $H = 5$, the comparison with the largest effect-size change; and 0.974 and 0.996 at checkpoint 2, $H = 10$. The two $H = 10$ comparisons therefore fall below 0.99 for the encoder-calibrated readout, and we make no claim that all four exceed it.

The same script measures how far the edit moves the latent, for one SAE feature direction over 20 of the contexts (the plan specified one direction per set; we measured only the feature set). Across the swept magnitudes the median displacement of the fifth rolled latent is 0.35% to 2.81% of its norm, but 1.8 to 14.8 in the whitened coordinates of Proposition 1, computed from all held-out encoded latents. By ordinary distance the edits are small; in the coordinates a readout is sensitive to they are not. This is consistent with the mechanism of §5.4 at intervention scale; we do not claim it establishes it.

Intervals on the effect sizes. Table 8 gives d as a point estimate, which cannot say whether a ratio between readouts is distinguishable from refitting noise. Resampling directions within each set, with the same indices applied to both readouts because the runs are paired, gives the intervals in Table 6. The ratio interval excludes 1 in 7 of the 12 contrasts and covers it in the remaining 5, which are all on checkpoint 1 and whose ratios run from 0.77 to 1.11. The 7 divide unevenly. Six are the checkpoint 2 contrasts, which are the ones scored over three variables rather than five (deviation (ii) above), so they are a replication on a restricted variable set. The single like-for-like five-variable contrast that survives is the pre-registered feature-versus-orthogonalised one on checkpoint 1 at $H = 5$, at 1.35 $[1.02, 1.89]$, and we read that as the result. Two of the 7 clear 1 only barely, with lower bounds 1.016

and 1.007; we apply no multiplicity correction across the 12, so the count is descriptive and those two would not survive one. The intervals are wide and asymmetric where C_{roll}'s d is small, since that d is the denominator. The resampling holds both readouts fixed and varies only which directions enter, so what it establishes is that the change is not direction-sampling noise; the within-calibration null on disjoint training halves, below, is what separates it from refitting noise.

Table 6: Paired bootstrap intervals on Cohen's d and on the ratio between readouts (10^4 resamples of directions, features and controls resampled within their own set and the same indices applied to both readouts, since the runs are paired). * marks a ratio interval excluding 1. The point estimates reproduce Table 8. Source: `data_runs/results/ic_cohens_d_bootstrap.json`, produced by `experiments/ic_frame/bootstrap_cohens_d.py`.

comparison	control set	d under C_{enc}	d under C_{roll}	ratio
checkpoint 1, $H = 5$	random	+1.083 [+0.709, +1.514]	+1.129 [+0.696, +1.687]	0.96 [0.67, 1.36]
	orthogonalised	+1.405 [+1.028, +1.859]	+1.044 [+0.640, +1.550]	1.35 [1.02, 1.89]*
	R^2-matched	+1.052 [+0.677, +1.478]	+0.999 [+0.588, +1.507]	1.05 [0.71, 1.63]
checkpoint 1, $H = 10$	random	+0.836 [+0.458, +1.270]	+1.089 [+0.683, +1.615]	0.77 [0.50, 1.05]
	orthogonalised	+1.243 [+0.863, +1.702]	+1.124 [+0.725, +1.628]	1.11 [0.85, 1.46]
	R^2-matched	+0.892 [+0.519, +1.324]	+1.030 [+0.626, +1.526]	0.87 [0.57, 1.26]
checkpoint 2, $H = 5$	random	+1.274 [+0.894, +1.726]	+0.532 [+0.155, +0.954]	2.39 [1.36, 7.16]*
	orthogonalised	+1.523 [+1.140, +1.990]	+0.754 [+0.395, +1.128]	2.02 [1.29, 3.86]*
	R^2-matched	+1.703 [+1.339, +2.165]	+0.664 [+0.302, +1.051]	2.57 [1.61, 5.57]*
checkpoint 2, $H = 10$	random	+1.425 [+1.002, +1.939]	+1.012 [+0.605, +1.512]	1.41 [1.01, 2.12]*
	orthogonalised	+1.837 [+1.420, +2.365]	+1.334 [+0.946, +1.808]	1.38 [1.04, 1.88]*
	R^2-matched	+1.844 [+1.446, +2.361]	+1.179 [+0.793, +1.645]	1.56 [1.17, 2.24]*

Magnitude is at the floor. IC is a direction statistic. Its companion, magnitude calibration $\text{MC} = 1 - \mathbb{E}|\,\|\Delta\hat{Q}\|/\|\Delta Q\| - 1\,|$ with the ratio clipped to $[0, 2]$, is stored per direction in `data_runs/ic_frame/` and we report it here. Under the rollout-calibrated readout on cartpole its mean is 0.000 to 0.001, with median 0.000, for all four direction sets at both $H = 5$ and $H = 10$, while mean feature IC is $+0.196$ and $+0.182$. The edits reproduce the direction of the physical change well enough to separate features from controls, and its magnitude not at all. Every IC statement in this paper is therefore about direction agreement, and we make no claim about the magnitude of an imagined intervention effect, which is the quantity Vakalis (2026) audits.

The encoder-calibrated readout scored here, and used for every δ of the sweep below, is the published file `data/readout.pt`. A first run of the sweep used `readout_C_enc.pt`, a closed-form refit of the same estimator whose weights differ along the simplicial group-sum direction, which the bias absorbs; on the IC latents the two agree to within 1.2×10^{-5} and per-direction IC agreed to within 1.3×10^{-5}. We re-ran those points with the published file so that the sweep matches its plan; the superseded runs are kept in `delta_sweep/superseded_refit_C_enc/`.

Intervention-magnitude sweep: exploratory, with a pre-specified analysis plan. The plan for this sweep was written and committed before any of its runs existed (`experiments/ic_frame/DELTA_SWEEP_PLAN.md`, commit `63fbf44`), which fixes the summaries reported here. It is nevertheless **exploratory**: it was decided after the main result was known, and it is not part of the pre-registration in Appendix C. We ran cartpole checkpoint 1 at $H = 5$ under both readouts at $\delta \in \{0.25, 0.5, 1.0, 2.0\}$, an eightfold range, holding the contexts, edits, controls, seed and every other setting at the published values. The $\delta = 1.0$ pair is the published comparison, reused rather than re-run, and this analysis reproduces its Cohen's d values exactly.

The effect is present at every magnitude and barely moves across the grid (Table 7). Mean feature IC runs 0.273 to 0.282 under the encoder-calibrated readout and 0.193 to 0.200 under the rollout-calibrated one; the largest ratio of $|d|$ between readouts is 1.34 to 1.35 across the four magnitudes, and the sign of feature-versus-control separation is preserved at every one. Feature IC stays rank-consistent between readouts throughout (Spearman ρ 0.872 to 0.876). Following the plan, we conclude only that the effect is not specific to the published magnitude on this checkpoint and horizon.

Table 7: Exploratory intervention-magnitude sweep, cartpole checkpoint 1 at $H = 5$. Cohen's d of SAE features against each control set under each readout, and their ratio. $\delta = 1.0$ is the published comparison. Source: `data_runs/results/ic_delta_sweep.json`.

	vs random		vs orthogonalised		vs probe-R^2-matched	
δ	C_{enc}	C_{roll}	C_{enc}	C_{roll}	C_{enc}	C_{roll}
0.25	1.07	1.11	1.39	1.03	1.04	0.98
0.5	1.07	1.12	1.39	1.03	1.05	0.99
1.0	1.08	1.13	1.40	1.04	1.05	1.00
2.0	1.11	1.15	1.43	1.07	1.06	1.02

Four points on one task and one checkpoint cannot establish magnitude independence, and we do not claim it.

Counts of directions whose per-direction bootstrap interval excludes zero, positive/negative, encoder-calibrated → rollout-calibrated readout, seed 1, $H = 5$: features 38/1 → 37/3; random 29/10 → 15/15, with mean random IC $+0.069 \rightarrow +0.006$. Seed 2, $H = 5$: features 41/0 → 28/2. Every per-direction value is in `data_runs/ic_frame/`; the encoder-calibrated values are the unmodified outputs of the existing analysis.

Per-direction conclusions, against refitting noise (post hoc). Each direction's bootstrap interval gives a call: significantly positive, significantly negative, or covering zero. `experiments/ic_frame/significance_flips.py` counts directions whose call differs between two readouts, using only existing runs. This count was not among the pre-registered outcomes, which were Spearman ρ and mean absolute change, so it is a post hoc analysis of pre-registered runs. In the published comparison (checkpoint 1, $H = 5$) the call differs for 9 of 50 features and 26 to 31 of 50 directions in each control set, and 7 to 12 control directions per set move between significantly positive and significantly negative; no feature does (0). The within-calibration null supplies the noise baseline: refitting one estimator on disjoint training halves changes 1 to 5 feature calls, 7 to 20 control calls per set and reverses 0 to 3, while swapping calibration source on the same half changes 7 to 9, 26 to 34 and reverses 8 to 12. The random control shows the pattern most plainly: under the encoder-calibrated readout 29 random directions are significantly positive and 10 significantly negative, under the rollout-calibrated readout 15 and 15, so whether random directions appear to carry positive IC depends on the calibration. Calls near the interval boundary can move with small shifts, which is why we compare against the within-source baseline and report reversals separately.

Table 8: Paired IC comparison on cartpole, encoder-calibrated → rollout-calibrated readout. ρ: Spearman correlation of per-feature IC between the two readouts, with 95% bootstrap interval. d: Cohen's d of the 50 SAE features against each of the three control sets. Last column: mean feature IC. Signs are preserved in all four rows; magnitudes are not. The rows are not like-for-like: checkpoint 1 scores IC over five physical variables and checkpoint 2 over three, because checkpoint 2's encoder-calibrated readout fails the pipeline's own gate on cart position and cart velocity (Appendix C), and the edit magnitude is $\delta = 1.0$ at $H = 5$ against $\delta = 3.0$ at $H = 10$. So d is computed on different vectors across rows, and the maximum over the four is not a maximum over replicates.

	feature ρ	d vs random	d vs orthogonalised	d vs R^2-matched	mean IC
checkpoint 1, $H = 5$	0.876 [0.766, 0.936]	$+1.083 \rightarrow +1.129$	$+1.405 \rightarrow +1.044$	$+1.052 \rightarrow +0.999$	$+0.277 \rightarrow +0.196$
checkpoint 1, $H = 10$	0.897 [0.796, 0.948]	$+0.836 \rightarrow +1.088$	$+1.243 \rightarrow +1.124$	$+0.892 \rightarrow +1.030$	$+0.232 \rightarrow +0.182$
checkpoint 2, $H = 5$	0.588 [0.373, 0.745]	$+1.274 \rightarrow +0.532$	$+1.523 \rightarrow +0.754$	$+1.703 \rightarrow +0.664$	$+0.280 \rightarrow +0.126$
checkpoint 2, $H = 10$	0.841 [0.717, 0.917]	$+1.425 \rightarrow +1.011$	$+1.837 \rightarrow +1.334$	$+1.844 \rightarrow +1.179$	$+0.277 \rightarrow +0.167$

Episode-level bootstrap. Table 9 gives 95% percentile intervals for the headline perturbed-cheetah quantities at $k = 5$, from 10,000 resamples of the held-out *episodes* of each checkpoint (`experiments/fidelity/bootstrap_ci.py`). Windows inside an episode overlap and are correlated, so resampling episodes rather than windows avoids understating uncertainty. The readouts are fit once on the training episodes and only the held-out set is resampled, so these in-

tervals describe held-out episode sampling within one checkpoint; model-to-model variation is the range across checkpoints reported everywhere else. The point estimates reproduce the values in Table 3 exactly.

Table 9: Episode-level bootstrap, perturbed cheetah at $k = 5$: point estimate with 95% percentile interval over 10,000 resamples of held-out episodes, per checkpoint. The encoder-calibrated interval is disjoint from both the rollout-calibrated and the realigned interval on every checkpoint; those two recovery intervals overlap with each other on every checkpoint.

quantity	checkpoint 1	checkpoint 2	checkpoint 3
encoder-calibrated	0.076 [−0.104, 0.231]	0.088 [−0.066, 0.221]	−0.049 [−0.250, 0.116]
rollout-calibrated	0.848 [0.839, 0.856]	0.832 [0.820, 0.843]	0.837 [0.823, 0.848]
realigned	0.843 [0.833, 0.851]	0.813 [0.798, 0.827]	0.814 [0.798, 0.827]
transfer gap	0.772 [0.623, 0.945]	0.745 [0.620, 0.888]	0.885 [0.729, 1.076]
held-out episodes	200	200	200
held-out windows	9000	9000	9000

Within-calibration null. Pre-registered in `experiments/ic_frame/PREREGISTRATION_NULL.md` after the first comparison and before any readout below existed. Training episodes were split into disjoint halves A and B; each readout estimator was fit on each half, and IC was run four times with everything else as in the seed 1, $H = 5$ analysis. All four readouts kept all five variables under the rolled-R^2 gate. Table 10 gives Spearman ρ of per-direction IC and the mean absolute change. Both pre-registered predictions held: feature ρ was higher within a calibration source than across sources, and feature IC changed less within a source than across.

Table 10: Within-calibration null (cartpole seed 1, $H = 5$). Spearman ρ of per-direction IC between two readouts, with 95% bootstrap interval, and mean absolute per-feature IC change.

readouts compared	features	random	orthogonalised	R^2-matched	feature $\lvert\Delta\text{IC}\rvert$
encoder-calibrated, A vs B	0.991 [0.974, 0.995]	0.926 [0.843, 0.965]	0.917 [0.802, 0.976]	0.876 [0.719, 0.964]	0.017
rollout-calibrated, A vs B	0.960 [0.893, 0.987]	0.673 [0.395, 0.869]	0.870 [0.720, 0.957]	0.861 [0.693, 0.958]	0.030
encoder- vs rollout-calibrated, half A	0.905 [0.823, 0.948]	0.126 [−0.195, 0.434]	0.256 [−0.090, 0.567]	0.196 [−0.140, 0.508]	0.089
encoder- vs rollout-calibrated, half B	0.884 [0.786, 0.938]	0.181 [−0.145, 0.485]	0.238 [−0.119, 0.557]	0.036 [−0.298, 0.372]	0.102

Cheetah diagnostic. On perturbed cheetah states (seed 1, $H = 5$) we trained a sparse autoencoder and per-feature probes as in the cartpole analysis and ran IC under both readouts with all 19 variables kept, since the encoder-calibrated readout passes the rolled-R^2 gate for 0 variables (mean rolled R^2 at $k = 5$: 0.145) and the rollout-calibrated readout for 2 (0.846). The best single-feature probe reached $R^2 = 0.579$, and 58% of dictionary features never fired. Under both readouts the top-50 features were not separated from controls: Cohen's d ranged from -0.026 to $+0.270$, and mean feature IC was $+0.017$ under the encoder-calibrated readout and $+0.018$ under the rollout-calibrated readout (Spearman $\rho = 0.776$ [0.590, 0.894], mean absolute change 0.025). We read this as a failure of the feature pipeline on this task, not as evidence about readout calibration, and we make no claim about cheetah's internal features.

The same paired bootstrap used in Table 6 makes the null precise rather than leaving it as an absence. Resampling directions with the same indices applied to both readouts, the feature-versus-control d is -0.026 $[-0.452, +0.345]$ against random directions, $+0.270$ $[-0.126, +0.632]$ against orthogonalised ones and $+0.136$ $[-0.271, +0.503]$ against R^2-matched ones under the encoder-calibrated readout, and $+0.094$, $+0.199$ and $+0.135$ under the rollout-calibrated one, every interval covering zero. The difference between the two readouts covers zero in all three (-0.120 $[-0.371, +0.101]$, $+0.071$ $[-0.142, +0.289]$, $+0.001$ $[-0.135, +0.160]$). So on the task with the largest transfer gap there is no separation for a calibration change to move, under either readout, and we cannot use cheetah to show the collapse and its mechanistic consequence in one cell. That cell is the experiment this paper most needs and does not have: it requires a task with both a large transfer gap and a sparse autoencoder that passes the pipeline's own design checks, and on our three tasks no such pairing exists. Reproduce with `experiments/ic_frame/cheetah_ic_bootstrap.py`.

G CORRECTIONS TO OUR OWN EARLIER INTERPRETATIONS

This work began as an attempt to replicate an IC analysis on walker and cheetah, and several interpretations we recorded along the way were wrong. We list them because each was plausible and each would have produced a different paper.

"Rolled cheetah latents have lost the physical state." On perturbed cheetah states the encoder-calibrated readout kept zero of 19 variables under the rolled-R^2 gate, and we initially read this as the dynamics discarding physical information. The diagnostics in §5.2 show that the measured physical variables remain linearly recoverable from rolled latents on this task and distribution, and that a target-agnostic linear realignment restores most of the original readout's accuracy within that condition. The picture is task dependent: on perturbed walker states the same realignment recovers far less (§5.2, and the small walker gain already noted in §E). §5.3 shows that this realignment is specific to its condition and is not well described by a rotation, so it does not establish a fixed coordinate relationship between the two representations.

"High cosine coexists with the collapse." We contrasted a latent cosine of 0.996 at $k = 5$ with the readout collapse. The cosine was measured on the planner's distribution and the collapse on perturbed states. On the perturbed distribution the cosine at $k = 5$ is 0.947, still above 0.9, so the qualitative point survives, but the original contrast mixed two distributions.

"The walker gate failure is explained by action dimensionality." A pre-registered cosine gate on wide-initial-state, random-action rollouts failed for walker (0.860 at $k = 5$) and passed for cheetah (0.958), although both have six action dimensions. Under the readout diagnostics of this paper, cheetah, which passed the cosine gate, is the task whose readouts fail most severely.

"Standardising targets will rescue the cheetah readout." Per-variable target standardisation was pre-registered and measured. For a shared ridge system solved independently per target it is an algebraic identity and changed nothing.

"The policy prior is on-policy." We first treated rollouts driven by the policy network as the model's own distribution. Transfer gaps under the planner are smaller than under the policy prior in all 15 walker pairs of checkpoint and horizon but in only 11 of 15 cheetah pairs, so the policy prior is itself a mild distribution shift, though not uniformly (Table 2).

H DREAMERV3 FEASIBILITY

We tested whether the analysis could be repeated on DreamerV3 (Hafner et al., 2023), using a public PyTorch reimplementation (`dreamerv3-torch`, commit `6ef8646`), on `dmc_walker_walk`. No released checkpoint was available, so a model would have to be trained. A smoke test built the configuration used for proprioceptive control tasks, collected random prefill through the repository's own simulation loop, constructed the agent (16,469,784 world-model parameters), and ran the encoder and recurrent state-space model on a real batch. The recurrent state has a 512-dimensional deterministic part h and a stochastic part of 32 categorical variables with 32 classes each, which is exactly one-hot, so an additive edit can only target h. An edit of norm 1.0 to h propagated through 5 imagined steps to the decoded observation.

Training one is the obstacle. The schedule for this configuration is 500,000 environment frames, or 125,000 updates at its training ratio, which on our hardware is 42.5 GPU-hours for the updates alone, excluding environment stepping, action selection and evaluation. That is 14 times the compute available to us per environment, so we did not train DreamerV3 models and the cross-architecture question is left open. Unrelated processes shared the machine during timing, so the per-update figure may be pessimistic, but closing the gap would need updates more than 14 times faster.

I SECOND ARCHITECTURE: LEWORLDMODEL ON REACHER

Choice of model. We looked for a pretrained latent world model with a separate encoder and learned latent dynamics, on continuous control, without simplicial normalisation and needing no training. The official LeWorldModel checkpoint for DeepMind Control Reacher (Maes et al., 2026) (`huggingface.co/quentinll/lewm-reacher`, revision `62adae4`, MIT) meets

these conditions; its weights match the Hub's recorded hash and load strictly into the architecture of the authors' code (`lucas-maes/le-wm` at `8edfeb3`). A ViT-tiny encoder maps one 224×224 frame to a 192-dimensional embedding, and a transformer predictor, conditioned on actions and up to three frames of context, is trained to predict the encoder's embedding of the next frame, so encoded and rolled latents share a space by construction. Other candidates we found were uploads without provenance, Atari or manipulation models, or TD-MPC2 checkpoints.

Inputs. Preprocessing follows the authors' code: frames rendered from the environment the training data were collected in (`stable-worldmodel` 0.1.1), ImageNet normalisation, one model step per five environment steps, and five z-scored actions concatenated per step. The action statistics come from a training set of about 300 GB of raw frames that we did not download; its collection script samples actions uniformly from $[-1, 1]^2$, so we used mean 0 and standard deviation $1/\sqrt{3}$. With these values the median one-step prediction error is 0.076 of the latent norm, against 0.702 for predicting no change and 0.309 or 1.003 with raw or badly scaled actions, so the model runs as trained.

Design. The analysis was committed before its first run and is exploratory, not pre-registered. We collected 1000 episodes of the training distribution, uniform random actions for 200 environment steps, and split them 800/200 by episode. The rolled latent at $t + k$ is the predictor's output after k autoregressive steps from the encoded frames at $t - 2$, $t - 1$ and t with the logged actions. Targets are the cosine and sine of both joint angles and the fingertip position; joint velocities, not visible in a single frame, were excluded from the mean in advance (their encoded-latent R^2 is at most -0.002). The estimators are the TD-MPC2 ones, unchanged. Basic realignment was part of the core diagnostic. Additional geometry and map-family analyses were to be run only if the transfer gap exceeded 0.05.

Table 11: LeWorldModel on Reacher, training distribution: mean held-out R^2 over the six position variables, and the transfer gap with a 95% interval from 2000 bootstrap resamples of held-out episodes.

k	encoded	C_{enc} on rolled	C_{roll}	realigned	no dynamics	transfer gap
1	0.999	0.997	0.999	0.999	0.950	0.002 [0.002, 0.002]
2	0.999	0.996	0.999	0.999	0.886	0.003 [0.003, 0.003]
3	0.999	0.996	0.999	0.999	0.823	0.003 [0.003, 0.003]
5	0.999	0.995	0.999	0.999	0.709	0.003 [0.003, 0.004]
10	0.999	0.994	0.998	0.998	0.495	0.004 [0.004, 0.004]

Result. The gap is distinguishable from zero but at most 0.004, so the geometry and map-family analyses were not run (Table 11). Realignment, part of the core diagnostic, leaves the readout where the rollout-calibrated one already is, since there is nothing to recover. The rollouts are not trivial: the no-dynamics baseline falls to 0.495 by ten steps and the rolled latents differ from the encodings by 0.161 of their norm, with latent cosine 0.986.

A descriptive comparison (post hoc). One explanation would be that LeWorldModel's isotropy regulariser removes the low-variance directions in which, for TD-MPC2, errors damage the readout. The numbers do not support it. With the TD-MPC2 definitions, LeWorldModel's latents have effective rank 78.7 of 192 (TD-MPC2: 2.8 to 52.7 of 512 across its nine task and distribution conditions), and 127 directions still share the last 10% of variance. At $k = 5$ they hold 22% of rollout-error energy (TD-MPC2: 24% to 91%) and 79% of the readout's squared weight (67% to 100%). Walker on perturbed states has a similar profile (24% and 68%) and a transfer gap of 0.073 at $k = 5$, so tail concentration alone does not predict the gap, as §5.4 already cautions. We did not measure what does separate the two models.

What this does not show. The test covers one task, one checkpoint and the training distribution only. TD-MPC2's largest gaps were on perturbed states, and the analogous test for LeWorldModel was not run. The result therefore bounds the claim without attributing the difference to architecture, task or state distribution. It does not bear on the LeWorldModel results of Vakalis (2026), which concern Cheetah and imagined intervention effects.

J PROBE DRIFT AND LABEL-FREE REALIGNMENT ON OTHER SHIFT AXES

The correction we fit has close analogues where the shift comes from something other than a model rolling its own dynamics, and the geometry reported there matches ours.

Kirichenko et al. (2023) show that retraining only the last layer on frozen features restores accuracy under spurious correlation, which is the inference our refit readout makes: a failing head does not imply missing features. Chen et al. (2025) fit an affine map between two language models' residual spaces and read the result with an unchanged probe, the same construction as our M followed by C_{enc}, and Csiszárik et al. (2021) use stitching to compare representations.

Dingeto (2026) is the closest. Probes matched to a language model decay to near-chance within 1,500 continued-training steps, yet an affine map fitted on paired activations without labels restores them in every case tested; decomposing that map, a rotation recovers about half and a non-orthogonal correction of rank 16 to 64 recovers nearly all. Duan (2026) reports the same staleness and the same label-free repair at 2B and 7B parameters. Our §5.3 finds the same shape on a different axis: the best orthogonal map recovers 40% of the full map's gain and identity plus a rank-16 correction recovers 85%.

We therefore claim no novelty for the finding that a drifted probe is repairable by a target-agnostic low-rank correction that is not a rotation. What their axis does not have, and what §5.3 and §5.5 measure, is the dependence of that correction on the rollout condition: it decays with horizon distance, can do worse than no map on another state distribution, and the choice of calibration source moves a mechanistic intervention measurement.

K RANKING THE TRANSFER GAP WITHOUT TARGETS

Measuring the transfer gap needs $q(x)$: the true physical state is required to score either readout. An analyst auditing a rollout usually does not have it, which makes the advice in §6 hard to act on. We therefore ask whether a quantity needing no targets, the mean latent cosine between rolled and encoded latents, ranks conditions by how badly their readout transfers. All 130 conditions (task $\times$ checkpoint $\times$ state distribution $\times$ horizon) come from the released `fidelity_long.csv`; `experiments/fidelity/gap_triage.py` reproduces everything here and writes `data_runs/results/gap_triage.json`.

It ranks. Spearman correlation between latent cosine and the transfer gap is -0.808 $[-0.851, -0.743]$ over the 130 conditions. The ordering is not an artefact of both quantities growing with the horizon: horizon alone ranks the gap at $+0.462$, and conditioning on horizon leaves the cosine correlation essentially unchanged (-0.779, -0.746, -0.744, -0.750 and -0.754 at $k = 1, 2, 3, 5, 10$). Inside a fixed task and horizon, where only the checkpoint and the state distribution vary, the correlation is below -0.5 in all 15 slices, from -0.709 to -1.000.

It transfers to a held-out task. Fitting the one-variable predictor on two tasks and applying it to the third preserves the ordering: Spearman $+0.983$ on cartpole, $+0.771$ on cheetah and $+0.900$ on walker.

It does not certify. On cartpole and walker the mean absolute error of the predicted gap (0.096 and 0.310) exceeds the gap's own standard deviation across those conditions (0.018 and 0.054), so the fitted value carries no usable magnitude information there. This is the same conclusion §5.4 and Proposition 1 reach from the other direction, and the two together give the sharper statement: a cosine value cannot tell you how much a readout loses, but the cosine *ordering* tells you where to look first. An analyst without targets can triage, ranking conditions by cosine and spending validation effort where it is lowest, which is weaker than certification and is available when certification is not.

L REPRODUCTION

`scripts/run_extension_pipeline.sh` regenerates every table, figure and number in this paper from the stored rollouts and checkpoints: it collects any missing state distributions, runs the fidelity analysis per checkpoint, the bound and identifiability checks, aggregation, figures and the generated macros. It requires inputs we do not redistribute: the released TD-MPC2 checkpoints,

the two cartpole checkpoints we trained, the LeWorldModel checkpoint, and the cached rollouts. Given those, the aggregation, bound, geometry, figure and macro stages run from the released code alone; the DreamerV3 feasibility stage (Appendix H) needs a separate environment and is skipped by default. Every number in the text is a macro from `paper_ext/generated/numbers.tex`, produced by `experiments/aggregate/make_numbers.py`; a missing value renders as a bold placeholder. The existing cartpole IC artifacts are read, never written; their hashes are verified before and after.