INNOVATION SIGREG: REGULARIZING TRANSITION RESIDUALS FOR JEPA WORLD MODELS

Anonymous authors
Paper under double-blind review

ABSTRACT

Gaussian regularization helps joint-embedding predictive architectures (JEPAs) resist collapse, or loss of distinctions between inputs. Yet the marginal embedding distribution pooled over trajectories need not be Gaussian, so enforcing this target can compete with linear state decoding. *Innovation SIGReg* applies Sketched Isotropic Gaussian Regularization to persistence-adjusted transition residuals, with a weighted raw-embedding anchor to constrain location and scale. Linear–Gaussian dynamics motivate the target: true transition noise can remain Gaussian while state distributions change, although learned residuals need not recover that noise. On Reacher, a robot-arm benchmark, mean linear angle-decoding R^2 at temporal stride 64 rises from 0.537 ± 0.092 to 0.787 ± 0.125 across ten runs per method, and the weakest-coordinate score rises from 0.217 to 0.669. Normalization alone and rescaled temporal differences do not reproduce these gains at the tested settings. On Two-Room navigation, unguarded residual regularization produces both strong and poor planners. An anchor guard improves mean success over marginal SIGReg by 4.00 percentage points in a prespecified first-cohort comparison and by 5.67 points in an exploratory second-cohort comparison; fixed and scheduled anchor increases perform comparably in the first. Detaching current-batch statistics yields $97.50 \pm 1.00\%$ success versus $76.00 \pm 17.39\%$ with differentiable statistics, although three runs per configuration leave reliability unresolved. A separate Reacher ablation reverses this comparison: detachment reduces R^2 from 0.421 to 0.010, while detached running averages perform best. These findings show that both the regularization target and the handling of its statistics matter for downstream decoding and planning.

1 INTRODUCTION

World models support planning by predicting observations under candidate actions (Ha & Schmidhuber, 2018; Hafner et al., 2019). Joint-embedding predictive architectures (JEPAs) predict in representation space (Assran et al., 2025; Maes et al., 2026), requiring embeddings to preserve distinctions needed for action. Prediction loss can instead favor discarding these distinctions, reaching *collapse* when embeddings become constant or nearly constant across inputs.

LeJEPA counters collapse with Sketched Isotropic Gaussian Regularization (SIGReg), encouraging isotropic Gaussian embeddings (Balestriero & LeCun, 2025); LeWorldModel (LeWM) combines it with action-conditioned prediction (Maes et al., 2026). Yet the *marginal* embedding distribution, pooled over trajectories, depends on initialization, dynamics, and policy. Invertible affine transformations preserve non-Gaussianity, so marginal matching can compete with *linear decoding*, or state estimation by linear regression. LeJEPA's distributional assumptions (Klindt et al., 2026) leave the best target on finite trajectories open.

We instead account for *persistence*, each coordinate's carry-over between sampled time steps. Innovation SIGReg regularizes residuals obtained by subtracting estimated carry-over from standardized embeddings and dividing by the residual scale implied by a linear–Gaussian model (Section 3.2). This model motivates the target without implying recovery of true transition noise (*innovations*). A weighted raw-embedding *anchor* constrains location and scale left unconstrained by residual standardization. We also test whether optimization differentiates through estimated statistics or treats them as constants (*detachment*).

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.

We evaluate Reacher angle decoding and Two-Room navigation. At Reacher stride 64, development-selected configurations, frozen before evaluation, improve mean and weakest-coordinate decoding across ten runs per method. Two-Room produces both strong and poor unguarded planners. An anchor guard improves mean success over marginal SIGReg in two separately specified experiment groups (*cohorts*), with prespecified and exploratory contrasts, respectively; fixed and scheduled anchors perform comparably in the first. Current-batch detachment gives the highest observed second-cohort mean across three models, but Reacher reverses this comparison and favors detached running averages. We make three contributions.

1. We formulate a persistence-adjusted residual regularizer with a raw-embedding anchor, and establish the scope and limitations of its linear–Gaussian motivation (Section 3).
2. We show gains in both average and weakest-coordinate Reacher decoding, and evaluate anchor interventions for planning in two separately reported cohorts (Sections 4.2–4.3).
3. We separate how statistics are updated from whether gradients pass through them, and document benchmark-dependent effects alongside the limits of variance concentration, state decoding, and training loss as measures of planning utility (Sections 4.4–4.7).

2 Background, related work, and problem formulation

2.1 JEPA world models and distributional regularization

Let $e_t = f_\theta(x_t) \in \mathbb{R}^d$ denote the embedding of observation x_t, with θ parameterizing the encoder and optional projector. JEPAs predict representations from related observations (Assran et al., 2023; Bardes et al., 2024). LeWM predicts future embeddings from observation and action (a_t) history; the Reacher setup instead minimizes $\|e_t - e_{t+\delta}\|_2^2$ without a learned predictor (Klindt et al., 2026). Here δ counts environment steps and $\mathcal{L}_{\text{pred}}$ denotes either loss. SIGReg matches random one-dimensional projections to a standard Gaussian, targeting zero mean, identity covariance, and Gaussian shape (Balestriero & LeCun, 2025); *marginal SIGReg* applies it directly to embeddings.

The regularized quantity distinguishes several approaches. VICReg constrains coordinate variance and covariance (Bardes et al., 2022); Barlow Twins targets cross-correlation between paired views (Zbontar et al., 2021); W-MSE whitens batch features before alignment (Ermolov et al., 2021). Related normalization methods whiten intermediate activations (Huang et al., 2018). These statistical constraints concern second moments rather than matching the complete Gaussian distribution. Sub-JEPA applies Gaussian regularization within multiple random embedding subspaces (Zhao et al., 2026), retaining a marginal target. Innovation SIGReg instead combines residual regularization with a weighted marginal term, the raw-embedding anchor.

2.2 Temporal structure and representation recovery

Slow Feature Analysis learns slowly varying features under variance and decorrelation constraints (Wiskott & Sejnowski, 2002), providing an early formulation of temporal coherence as a representation-learning objective. TC-LeWM subtracts a local temporal mean before applying SIGReg and evaluates representations through robot policy learning (Liu et al., 2026). Both methods therefore regularize temporal residuals. Our distinction is to subtract estimated coordinatewise persistence and divide by the residual standard deviation predicted by a linear transition model (Section 3.2). We compare this with *normalization alone*, which regularizes standardized next embeddings, and *temporal differencing*, which regularizes differences between standardized embeddings after rescaling each difference coordinate. Our Two-Room comparison also adapts TC-LeWM's regularizer to the shared planning pipeline.

Temporal structure can support *identification*: recovering hidden variables up to transformations allowed by a generative model. Time-contrastive learning exploits distributional changes over time (Hyvarinen & Morioka, 2016), while permutation-contrastive learning exploits temporal dependence in stationary sources (Hyvärinen & Morioka, 2017). Independent Innovation Analysis uses mutually independent innovation components whose distributions vary with observed auxiliary information (Morioka et al., 2021). LeJEPA's recovery results likewise depend on its specified stationary model class, with time-invariant state distributions (Klindt et al., 2026). These guarantees do not transfer

automatically to estimated-residual penalties. At matched dimension, *linear state recovery* requires an invertible affine relationship between embeddings and state. A *linear probe* instead fits a regression from fixed embeddings to selected state variables; its decoding score measures their accessibility under the evaluation protocol, not general state or innovation recovery.

2.3 THE TWO DESIGN QUESTIONS

We ask which distribution to regularize and how to update and differentiate through its statistics. Comparisons use linear decoding and planning, retaining each benchmark's architecture and evaluation protocol. Section 4.1 specifies hyperparameter selection and cohort design; Appendix I expands the literature comparison.

3 METHOD: INNOVATION SIGREG

3.1 WHY REGULARIZE TRANSITION INNOVATIONS?

An *episode* is a finite trajectory generated from an initial state. Episode initialization can change the distribution of visited states without changing transition noise. To isolate this distinction, consider a latent state $s_t \in \mathbb{R}^n$ following a diagonal first-order autoregressive (AR(1)) model, where each coordinate depends linearly on its previous value:

$$s_{t+\delta} = \rho_\delta \odot s_t + \sqrt{\mathbf{1} - \rho_\delta^2} \odot \varepsilon_t, \qquad \varepsilon_t \sim \mathcal{N}(0, I_n), \quad \varepsilon_t \perp s_t. \tag{1}$$

Here $\rho_{\delta,j}$ is coordinate j's persistence coefficient, $|\rho_{\delta,j}| < 1$, $\mathbf{1}$ is the all-ones vector, and I_n is the identity matrix. Products, powers, and roots act coordinatewise; $\perp$ denotes independence. Subtracting the persistent component and dividing by $\sqrt{\mathbf{1} - \rho_\delta^2}$ recovers the standard Gaussian innovation ε_t for any initial-state distribution (Appendix Lemma 1). For example, Gaussian states whose variances change across an episode can form a non-Gaussian pooled distribution, although their innovations have the same Gaussian distribution.

This motivation concerns transient state distributions. At stationarity, when the state distribution no longer changes over time, the same diagonal transition and noise scaling with $|\rho_{\delta,j}| < 1$ and state-independent innovations has a standard Gaussian state distribution if and only if its innovations are standard Gaussian (Appendix Proposition 1). Thus the argument does not justify Gaussian innovations for a stationary non-Gaussian state distribution. Nor do we assume that learned embeddings satisfy this model or that their residuals recover its innovations.

3.2 CONSTRUCTING THE REGULARIZED RESIDUAL

Reacher trains on observation pairs $(x_t, x_{t+\delta})$; LeWM trains on four-frame sequences and forms residuals between adjacent sampled frames. Pairs never cross episode boundaries. We estimate embedding means and variances from Reacher's source observations x_t, or from all LeWM sequence frames, and use the same moments to standardize both members of each pair. Averaging their coordinatewise standardized products estimates persistence (Appendix B.1). A star denotes the statistics chosen for the current update: mean μ^*, variance v^*, and persistence estimate $\widehat{\rho}^*$, taken from either the current batch or running averages (Section 3.3). With $e_t = f_\theta(x_t)$ denoting the raw embedding, we compute its standardized version z_t and the transition residual η_t:

$$z_t = (e_t - \mu^*) \oslash \sqrt{\max(v^*, \epsilon_v)}, \qquad \rho^* = \mathrm{clip}(\widehat{\rho}^*, -\rho_{\max}, \rho_{\max}), \tag{2}$$

$$\eta_t = (z_{t+\delta} - \rho^* \odot z_t) \oslash \sqrt{\mathbf{1} - (\rho^*)^2}. \tag{3}$$

Here $\oslash$ denotes coordinatewise division, $\epsilon_v = 10^{-8}$ is a numerical variance floor, and $\rho_{\max} = 0.995$ bounds the coefficient magnitude. The numerator subtracts estimated persistence: it approaches a temporal difference as ρ^* approaches one, and the standardized next embedding as ρ^* approaches zero. The denominator is the residual standard deviation implied by a stationary, unit-variance AR(1) model; clipping limits its amplification. Source and next observations can have different means and variances, so applying common normalizers does not ensure a zero-mean, unit-variance residual,

even with an exact coefficient (Appendix A.3). SIGReg therefore penalizes moment errors as well as departures from Gaussian shape.

We apply SIGReg to η (Appendix B.3). Its persistence estimate is separate from the world model's predictor and is not conditioned on actions: the residual can include action effects and need not equal prediction error. Matching its distribution also does not enforce independence from states or actions. The objective is

$$\mathcal{L} = \alpha_\lambda \mathcal{L}_{\text{pred}} + \lambda\big[\,\text{SIGReg}(\eta) + w_{\text{m}}\,\text{SIGReg}(e)\big], \tag{4}$$

where λ controls regularization strength and w_{m} is the raw-embedding anchor weight relative to the residual term. We preserve each baseline's loss convention: $\alpha_\lambda = 1$ for LeWM and $1 - \lambda$ for Reacher. Because the pipelines aggregate losses differently, these weights are not directly comparable across benchmarks (Appendix B.3).

The anchor constrains changes that residual normalization cannot detect. With batch statistics and an inactive variance floor, coordinatewise shifts and positive rescalings of e leave η unchanged. Shrinking embeddings can therefore reduce alignment loss toward collapse without changing the residual penalty. The anchor $\text{SIGReg}(e)$ remains sensitive to raw location and scale; w_{m} balances this constraint against residual matching, without guaranteeing that every collapsed solution is excluded. Figure 1 summarizes the computation.

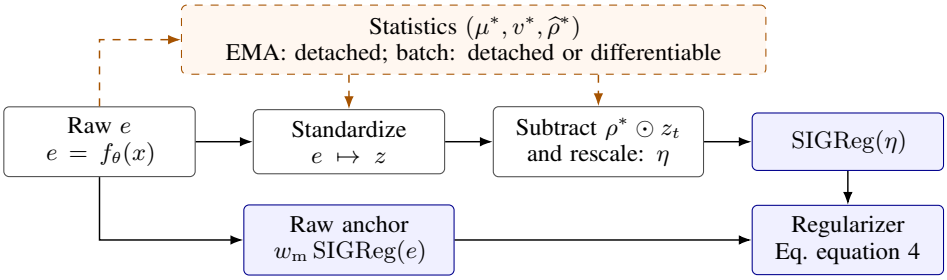

Figure 1: Innovation SIGReg. Solid paths retain gradients through the embeddings. Dashed paths supply estimated statistics; their backward paths are blocked in the detached modes. The residual and raw-embedding anchor are combined with the original prediction or alignment loss in Eq. equation 4.

3.3 Updating statistics and choosing gradient paths

Statistics introduce two choices: how they are updated and whether gradients pass through them. Running averages smooth batch fluctuations but can lag behind a changing encoder; current-batch statistics respond immediately. We compare *EMA-detached* statistics, using exponential moving averages with update weight 0.01; *batch-detached* statistics, using current-batch estimates; and *batch-differentiable* statistics, using those same estimates with gradients retained. In both detached modes, backpropagation treats the selected statistics as constants. Comparing detached modes tests the statistics update; comparing batch modes tests the gradient path. We do not study differentiable running statistics.

After selecting statistics, we recompute z from the current encoder outputs, preserving direct gradients through both observations in each pair. At identical embeddings and random projections, the batch modes have equal losses but different gradients (Appendix Algorithm 1). Only the differentiable mode accounts for how an encoder change alters the estimated moments and persistence. Under the affine invariance above, differentiating these statistics makes the residual gradient orthogonal to coordinatewise shifts and rescalings; detachment need not preserve this property (Appendix B.5).

The optional Two-Room guard monitors covariance participation ratio (PR), a measure of how many directions carry substantial embedding variance, rather than algebraic rank. Falling PR signals concentration into fewer directions, motivating stronger raw-embedding regularization. When smoothed PR falls below 12, the guard raises w_{m} from 0.1 to 0.5 and retains that increase. A fixed-anchor control uses 0.5 throughout; a scheduled control makes the same increase at a predetermined update. These controls test whether the strength or timing of the anchor explains the benefit without adaptive triggering (Appendix B.6).

4 EXPERIMENTS AND ANALYSIS

4.1 EVALUATION DESIGN

We first test whether transition regularization preserves linearly accessible state information, then whether the resulting world model supports planning. The Reacher benchmark of Klindt et al. (2026) decodes two joint angles from a two-dimensional CNN embedding. The probe is fitted on trajectory observations and scored on 10,000 independently rendered Gaussian-angle configurations, measuring decoding under distribution shift. Two-Room evaluates action-conditioned planning with LeWorldModel (Maes et al., 2026), a 192-dimensional embedding, and its cross-entropy-method/model-predictive-control planner. The main planning cohorts share a 2,000-episode dataset regenerated with the upstream collector. Official-data and manipulation experiments provide separate transfer checks.

Primary benchmark comparisons use final checkpoints after 100 epochs and include every completed run in each cohort. Each model's planning success rate (SR, %) averages four evaluation seeds with 50 episodes each; tables report the mean, sample SD, and lowest model-level SR. Reacher's co-primary outcomes are the mean of the two coordinatewise R^2 scores and their minimum, $R^2_{\min}$. Reported Welch 95% intervals are post-hoc, unadjusted summaries under an independent-run approximation; Two-Room initialization and random-state independence remain unverified (Appendix C.3). Nominal seed labels are not treated as verified pairs.

Configurations were selected during development and frozen before each cohort; cohorts are never pooled. At Reacher stride 64, marginal and Innovation SIGReg retain their selected weights, $\lambda = 0.05$ and 0.005. Innovation received more development configurations, and its normalization and differencing controls reuse its selected setting. Two-Room uses $\lambda = 0.09$ throughout. Prespecified planning contrasts were guard versus marginal, Sub-JEPA, and scheduled anchor in the first cohort; the second specified batch-detached versus batch-differentiable innovation and the three pairwise comparisons among guard, differencing, and TC-LeWM. The first cohort also specified secondary contrasts of unguarded innovation versus normalization-only and marginal SIGReg, and of guard versus fixed anchor. Other contrasts, including second-cohort guard versus marginal, are exploratory. Appendix C details selection, uncertainty estimates, and deviations from the protocols.

4.2 TRANSITION RESIDUALS IMPROVE LINEAR ANGLE DECODING

At temporal stride $\delta = 64$, Innovation SIGReg raises mean decoding R^2 from 0.537 ± 0.092 to 0.787 ± 0.125 (Table 1). The gain is 0.251 (95% interval $[0.147, 0.355]$), while the weakest-coordinate score increases from 0.217 to 0.669, a gain of 0.451 ($[0.242, 0.661]$). Thus the improvement extends to the less accessible angle: nine of ten innovation runs exceed $R^2 = 0.3$ on both coordinates, compared with three of ten marginal runs.

Table 1: **Fresh Reacher comparison.** All-run mean ± sample SD. δ is temporal stride, n counts trained models, and "Both" counts models with $R^2_{\min} > 0.3$. Innovation and the stride-64 controls use EMA-detached statistics. Stride 16 uses a separately selected innovation setting ($\lambda = 0.01$, $w_{\mathrm{m}} = 1$); it is not a stride-only intervention.

δ	Regularizer	n	Overall R^2	$R^2_{\min}$	Both
64	Marginal SIGReg	10	0.537 ± 0.092	0.217 ± 0.185	3/10
64	Innovation SIGReg	10	$\mathbf{0.787 \pm 0.125}$	$\mathbf{0.669 \pm 0.252}$	**9/10**
64	Normalization only	10	0.502 ± 0.102	0.178 ± 0.192	1/10
64	Temporal difference	10	0.467 ± 0.004	0.023 ± 0.004	0/10
16	Marginal SIGReg	5	0.371 ± 0.152	0.112 ± 0.311	1/5
16	Innovation SIGReg	5	0.510 ± 0.108	0.259 ± 0.204	1/5

The controls help locate the benefit. Regularizing the standardized next embedding gives $R^2 = 0.502 \pm 0.102$; a standardized temporal difference gives 0.467 ± 0.004, but only 0.023 ± 0.004 on the weaker coordinate. The weakest-coordinate gains over these controls are 0.490 ($[0.279, 0.702]$) and 0.646 ($[0.466, 0.826]$), respectively. The estimated-persistence residual is therefore the better tested target at the selected innovation setting. These comparisons neither establish the controls' best

achievable performance after separate tuning nor isolate persistence subtraction from residual scaling. At $\delta = 16$, mean R^2 increases from 0.371 to 0.510, but both methods decode both angles above the threshold in only one of five runs. Development sweeps at strides 1 and 2 yield no such runs.

Development ablations clarify the stride-specific anchor choice (Table 9). At $\delta = 16$ and $\lambda = 0.005$, increasing w_{m} from 0.1 to 1 changes R^2 from 0.392 ± 0.015 to 0.618 ± 0.157 (three models each). At $\delta = 64$, the same weight change lowers the recorded mean from 0.755 ± 0.154 ($n = 10$) to 0.444 ± 0.011 ($n = 3$). With $w_{\mathrm{m}} = 1$ at $\delta = 16$ and $\lambda = 0.005$, the full-matrix transition and residual whitener reaches 0.565 ± 0.104 ($n = 3$), below the diagonal mean in that cell. These selected development comparisons show why stronger anchoring or a richer transition estimate cannot be assumed to improve decoding.

True-angle diagnostics offer relevant context: fitted residual excess kurtosis changes from -0.56 at stride 1 to -0.043 and -0.053 at strides 16 and 64. Yet the pooled estimator retains residual–state correlation of 0.208 at stride 64; near-zero kurtosis does not establish independent noise or explain the decoding gain. A label-shuffling probe control gives $R^2 = 0.0003$. Wrapped-angle inputs score 0.8149, a linear-probe reference, not an information ceiling. Appendix D provides the individual runs and complete diagnostic definitions.

4.3 Planning gains depend on stabilization

Planning exposes a separate difficulty: a useful residual target can still train unreliably. In the first Two-Room cohort, unguarded batch-differentiable Innovation SIGReg reaches 97.5–99.5% SR in three runs but 58.0% and 50.5% in the other two (Table 2A). Its all-run mean, $80.70 \pm 24.30\%$, falls below marginal SIGReg's $88.70 \pm 0.84\%$. The anchor guard, triggered by a drop in covariance participation ratio (PR), raises the mean to $92.70 \pm 1.10\%$ and the lowest outcome to 91.0%. Its mean advantage over marginal SIGReg is 4.00 points (95% interval $[2.56, 5.44]$).

Table 2: **Two separately specified Two-Room cohorts.** SR is mean $\pm$ sample SD across n trained models; "Worst" is the lowest model-level SR, all in percent. I and J share the unguarded, batch-differentiable loss configuration; B detaches its current-batch statistics. T has three executed runs in the first cohort; the remaining two optional slots were not run. Appendix C defines all arms and Appendix E lists every evaluation.

A. First cohort

Regularizer / control	n	SR	Worst
M: Marginal	5	88.7 ± 0.8	88.0
I: Innovation	5	80.7 ± 24.3	50.5
H: Fixed anchor	5	91.9 ± 2.0	90.0
S: Schedule	5	91.4 ± 2.3	88.0
G: Guard	5	92.7 ± 1.1	91.0
Z: Normalize next	5	51.5 ± 1.3	50.0
U: Sub-JEPA	5	91.4 ± 1.0	90.0
T: Difference	3	97.8 ± 1.6	96.0

B. Second cohort

Regularizer / control	n	SR	Worst
M: Marginal	3	87.0 ± 1.8	85.5
G: Guard	3	92.7 ± 1.6	91.5
T: Difference	3	85.0 ± 23.0	58.5
C: TC-LeWM	3	51.8 ± 20.1	31.0
J: Batch-diff.	3	76.0 ± 17.4	57.5
B: Batch-detached	3	$\mathbf{97.5 \pm 1.0}$	96.5

A stronger fixed anchor, a scheduled anchor increase, and Sub-JEPA (Zhao et al., 2026) reach 91.90%, 91.40%, and 91.40%. Their gaps to the guard are at most 1.3 points, with intervals including zero; the data do not establish a benefit from adaptive triggering over these controls. The schedule switches at update 4,960, selected from six earlier trigger times spanning 4,183–5,703. The guard triggers at 3,978–5,469 in this cohort, placing the stronger anchor at a similar training stage (Appendix E.4). Normalization alone reaches $51.50 \pm 1.27\%$ with final raw-embedding PR of 1.0 in every run, so standardization is insufficient in this setting.

In an exploratory comparison, the second cohort again favors the guard over marginal SIGReg: $92.67 \pm 1.61\%$ versus $87.00 \pm 1.80\%$, a 5.67-point difference ($[1.78, 9.56]$). Temporal differencing, initially strongest at $97.83 \pm 1.61\%$, falls to $85.00 \pm 22.96\%$ with one 58.5% run. This reversal cautions against inferring reliability from three strong outcomes. The four-frame TC-LeWM adaptation (Liu et al., 2026) reaches $51.83 \pm 20.05\%$ at the shared $\lambda = 0.09$, without an anchor or separate tuning. This evaluates its regularizer within the Two-Room planner; the original LIBERO policy-learning system is outside the comparison.

4.4 THE EFFECT OF STATISTICS HANDLING DEPENDS ON THE BENCHMARK

The batch-detached (B) and batch-differentiable (J) variants isolate a specific computational choice: the backward path through current-batch mean, variance, and lag estimates. At fixed inputs, detachment preserves the forward residual and loss while retaining direct embedding gradients; an implementation check confirms loss agreement to within 10^{-10}. B reaches 98.5, 97.5, and 96.5% SR, whereas J reaches 57.5, 78.5, and 92.0% (Figure 2a). The prespecified B–J mean difference is 21.50 points, with a wide 95% interval of $[-21.49, 64.49]$. B's exploratory advantage over marginal SIGReg is 10.50 points ($[6.80, 14.20]$). These outcomes are consistent with an optimization benefit from detaching statistics, but three runs per arm and unverified initialization matching leave its reliability unresolved. The caution prompted by the initial three strong differencing runs applies equally to B's three outcomes. Matching nominal labels do not establish paired effects.

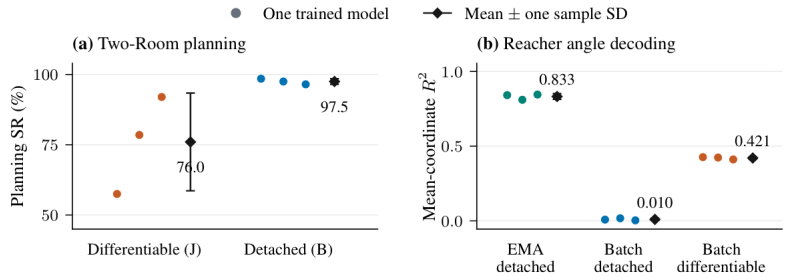

Figure 2: **Statistics choices have benchmark-dependent effects.** Circles show individual models; black diamonds and bars show the mean and sample SD ($n = 3$ per configuration). Numeric annotations are means. (a) Two-Room, second cohort, using current-batch statistics. (b) Reacher at $\delta = 64$; EMA (exponential moving average) uses running estimates. Differentiable variants pass gradients through the statistics; detached variants treat them as constants. Horizontal offsets separate points and imply no pairing.

Reacher reverses the current-batch comparison (Figure 2b): batch detachment reduces mean R^2 from 0.421 ± 0.008 to 0.010 ± 0.007, while EMA-detached statistics perform best at 0.833 ± 0.019. Historical EMA-detached Two-Room runs score 79.0, 96.0, and 79.0%; they provide context but are not a matched test of statistics freshness. Thus the two benchmarks favor different statistics choices. Architecture, embedding dimension, sequence sampling, and objective details vary together, preventing attribution of the reversal to any one of them.

4.5 REPRESENTATION MEASUREMENTS AND PLANNING

We next ask whether representation geometry and position accessibility account for the planning differences. Figure 3a,b aligns each model's planning SR, raw PR, and mean-coordinate position-decoding R^2 across all 56 models, with the first and second cohorts shown separately. PR measures how variance is distributed across covariance directions, not the number of nonzero dimensions. In the second cohort, B achieves the highest SR at PR 2.3–2.9, compared with 14.5–15.3 for the guard and 29.5–30.3 for marginal SIGReg. TC-LeWM's poorer outcomes span PR 2.3–15.0. Variance spread therefore does not order these methods.

Coordinate-wise position probes reveal more specific differences (Table 25). All three TC-LeWM models have negative R^2 on both coordinates. Second-cohort marginal models decode x well (0.90–0.95), but y inconsistently (-0.05–0.73), echoing Reacher's weaker-coordinate deficit. Nevertheless, B and J have similar mean position scores, 0.896 ± 0.041 and 0.890 ± 0.109, despite their planning gap. The 57.5% J model has the lowest B/J probe score, 0.765; more strikingly, a first-cohort normalization-only model scores 0.905 but plans at 51.0%. Accessible position therefore does not explain every outcome. The probes standardize each feature before fitting, so small-variance directions can support decoding even when raw PR is near one. These probes decode observations, not predicted rollouts or the planner's use of latent distances.

The early PR alarm has a related limitation. PR below 2 at epoch 10 flags all five first-cohort unguarded innovation runs, including the three that finish above 97% SR, and none of the marginal

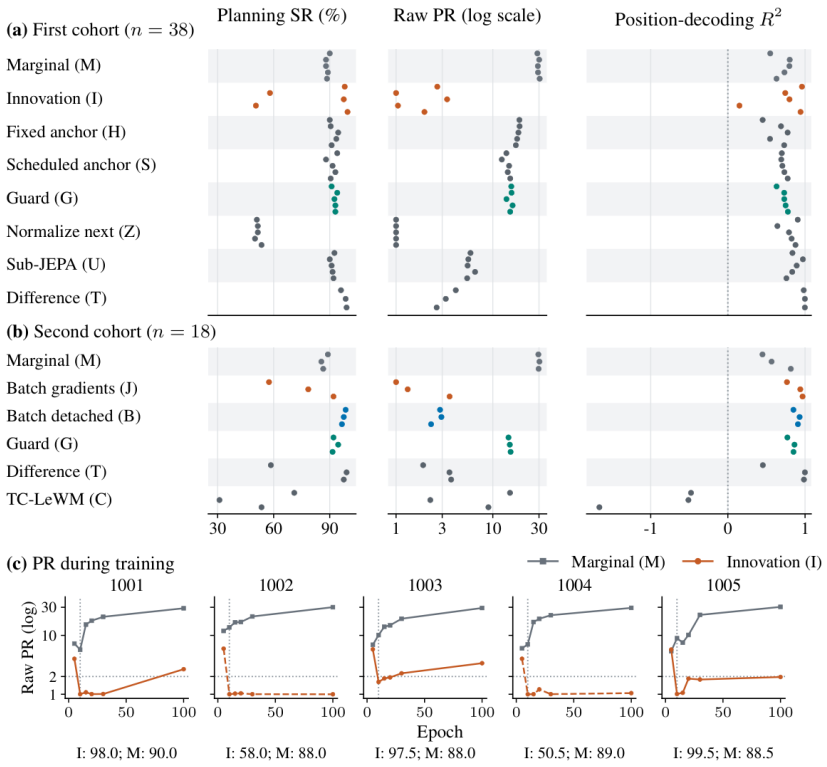

Figure 3: **Representation measurements and planning.** (a,b) All 56 models, grouped by cohort and method. Each model occupies the same vertical position across the SR, raw-PR, and mean-coordinate position-R^2 columns; within-row offsets only separate models, ordered by nominal run label. (c) First-cohort trajectories, split by nominal label; numbers below give final SR (%) for innovation (I) and marginal (M). Dotted guides mark epoch 10 and PR 2; dashed curves identify I labels 1002 and 1004. These are offline measurements, distinct from the online guard. The results appendix F specifies the common manifest and position-probe protocol.

runs (Figure 3c). Final PR is 1.00–1.05 in the two low-SR innovation runs, but a 99.5% planner also finishes at PR 1.97. Low prediction loss is similarly insufficient: the 50.5% planner's final validation prediction loss is 0.0048, below 0.0070 for the 98.0% planner. Final regularizer loss, by contrast, is highest for the two poor planners (2.71 and 2.63, versus 1.76–2.09), an observation from five runs that we did not test in other arms. The guard improves training outcomes in the tested cohorts, while these measurements do not certify eventual planning quality.

4.6 Persistence alone does not select the target

To examine persistence as an explanation, we vary the transition coefficient $\rho \in \{0.50, 0.89, 0.99\}$ in a two-dimensional autoregressive system with Gaussian transition noise, a fixed bimodal source distribution, and an invertible nonlinear observation map. The arms share architecture, optimizer, and training budget and are evaluated on independent two-state episodes (Appendix G). At $\rho = 0.89$, marginal SIGReg reaches $R^2 = 0.989 \pm 0.001$ with stationary Gaussian sources but 0.648 ± 0.284 with the bimodal source, confirming that the system supports near-perfect linear decoding when the marginal target matches the state distribution.

Under the bimodal source, innovation trails marginal SIGReg at $\rho = 0.50$ (0.839 ± 0.003 versus 0.951 ± 0.002). Its innovation-minus-marginal gaps at 0.89 and 0.99 are $+0.103$ and -0.159, with substantial variation across three runs per cell. Its gaps to differencing are also nonmonotone: $+0.021$, -0.075, and $+0.018$. Changing ρ changes the next-state marginal, and measured residual–state dependence remains above the permutation reference. This experiment tests the implemented

estimator, not recovery of the known generative noise, and provides no persistence-only rule for choosing a target.

A fixed-embedding experiment separates estimator behavior from encoder optimization (Table 28). At $\rho = 0.9$, a stationary Gaussian autoregression gives residual variance 0.992. Shuffling evaluation frames while retaining statistics fitted on intact trajectories gives variance 9.610 and residual–state correlation 0.672, despite near-zero excess kurtosis (-0.028). Nonstationary shifted and bimodal initial distributions instead give variances 0.671 and 0.786, even with Gaussian transition noise. Temporal correspondence and normalization therefore affect the estimated target before representation learning; near-zero kurtosis alone does not validate that target.

4.7 Transfer and scope of the findings

The earlier unguarded, batch-differentiable configuration remains below marginal SIGReg in all-run mean on official Two-Room data: $83.8 \pm 20.8\%$ versus $87.7 \pm 1.6\%$, with three models each. Its individual outcomes, 93.0, 60.0, and 98.5%, show that the poor-run pattern also occurs on official data. On PushT it reaches $79.0 \pm 7.5\%$ over three models, against a single marginal model at 86.0%. The batch-detached configuration has not been evaluated in either setting (Appendix E).

The innovation statistic itself is also an unreliable guide to decoding quality in our development experiments. Across 299 records its pooled Spearman correlation with R^2 is 0.034. Trajectory subgroups range from 0 to 0.50, whereas both Ornstein–Uhlenbeck subgroups are negative (Appendix G). These inconsistent associations reinforce the distinction between a useful training penalty and a diagnostic of what a representation has learned.

5 Discussion

Reacher's clearest gain is on the weaker coordinate: at stride 64, its improvement over marginal SIGReg (0.451) exceeds the mean-decoding gain (0.251). Normalization and differencing controls support the tested residual construction, though they were not separately tuned and do not isolate persistence subtraction from scaling. The gain remains stride-specific.

Two-Room establishes stabilization's value. The guard improves mean planning success over marginal SIGReg in both cohorts, but comparable first-cohort fixed and scheduled anchors leave an advantage from adaptive triggering unestablished. Differencing and Sub-JEPA leave the residual target's superiority unresolved. Strengthening the anchor rebalances marginal and residual penalties, yet reduces development decoding at Reacher stride 64, so its benefit requires evaluation within each benchmark.

Current-batch detachment yields the highest observed second-cohort planning mean, but Reacher reverses this comparison and favors detached running averages. Affine invariance explains gradient differences without selecting modes. The Two-Room comparison has three runs per configuration, with initialization matching and random-state independence unverified, leaving reliability unresolved.

Controlled experiments yield no persistence-only rule for choosing a target, and estimated residuals need not recover true innovations. PR, position decoding, and prediction loss do not individually certify planning quality. Transfer of the later batch-detached configuration to official Two-Room and manipulation data remains untested. Residual construction, anchor strength, and statistics handling therefore require evaluation through downstream decoding and planning.

6 Conclusion

Innovation SIGReg combines Gaussian regularization of persistence-adjusted transition residuals with a raw-embedding anchor. At Reacher stride 64, it improves mean and weakest-coordinate decoding over marginal SIGReg and the tested normalization and differencing controls. In Two-Room, the anchor guard improves mean planning success in both cohorts, although fixed and scheduled anchors perform comparably in the first. The current-batch detachment comparison reverses between benchmarks, showing that target construction and statistics handling require evaluation together. Whether combining batch detachment with the guard improves planning, and whether these configurations transfer to broader control tasks, remain open.

AI USE STATEMENT

We used generative AI tools to refine manuscript wording, assist with code debugging, and support selected aspects of methodological development, hypothesis refinement, proof preparation, and interpretation of results. The authors reviewed the AI-assisted material, checked the mathematical arguments and interpretations, and verified LLM-generated or modified code for correctness. All final research decisions and conclusions remained with the authors, who take responsibility for the manuscript and accompanying artifacts.

ETHICS STATEMENT

This work studies representation learning and planning using simulated environments and controlled synthetic experiments. It does not involve human-subject experiments or personal data. The reported improvements concern decoding and planning under the specified evaluation protocols; they do not establish the safety or reliability of deployment in physical systems. Applications to real-world autonomous systems would require additional validation under deployment conditions, including assessment of failure modes and appropriate safeguards.

REPRODUCIBILITY STATEMENT

Section 3 describes the proposed method, with mathematical assumptions and proofs in Appendix A and implementation details in Appendix B. Experimental settings, data collection, hyperparameter selection, and evaluation protocols are documented in Appendix C. Appendix H describes result provenance and the materials for regenerating tables and figures. An anonymized codebase accompanies the submission. Model checkpoints are omitted because of submission upload size restrictions. Upon acceptance, we will publicly release the codebase and all model checkpoints associated with the reported experiments.

REFERENCES

Mahmoud Assran, Quentin Duval, Ishan Misra, Piotr Bojanowski, Pascal Vincent, Michael Rabbat, Yann LeCun, and Nicolas Ballas. Self-supervised learning from images with a joint-embedding predictive architecture, 2023. URL https://arxiv.org/abs/2301.08243.

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

Randall Balestriero and Yann LeCun. Lejepa: Provable and scalable self-supervised learning without the heuristics, 2025. URL https://arxiv.org/abs/2511.08544.

Adrien Bardes, Jean Ponce, and Yann LeCun. Vicreg: Variance-invariance-covariance regularization for self-supervised learning, 2022. URL https://arxiv.org/abs/2105.04906.

Adrien Bardes, Quentin Garrido, Jean Ponce, Xinlei Chen, Michael Rabbat, Yann LeCun, Mahmoud Assran, and Nicolas Ballas. Revisiting feature prediction for learning visual representations from video, 2024. URL https://arxiv.org/abs/2404.08471.

Xinlei Chen and Kaiming He. Exploring simple siamese representation learning, 2020. URL https://arxiv.org/abs/2011.10566.

T. W. Epps and L. B. Pulley. A test for normality based on the empirical characteristic function. *Biometrika*, 70(3):723–726, 1983. doi: 10.1093/biomet/70.3.723. URL https://academic.oup.com/biomet/article-abstract/70/3/723/248040.

Aleksandr Ermolov, Aliaksandr Siarohin, Enver Sangineto, and Nicu Sebe. Whitening for self-supervised representation learning, 2021. URL https://arxiv.org/abs/2007.06346.

Quentin Garrido, Randall Balestriero, Laurent Najman, and Yann Lecun. Rankme: Assessing the downstream performance of pretrained self-supervised representations by their rank, 2023. URL https://arxiv.org/abs/2210.02885.

Jean-Bastien Grill, Florian Strub, Florent Altché, Corentin Tallec, Pierre H. Richemond, Elena Buchatskaya, Carl Doersch, Bernardo Avila Pires, Zhaohan Daniel Guo, Mohammad Gheshlaghi Azar, Bilal Piot, Koray Kavukcuoglu, Rémi Munos, and Michal Valko. Bootstrap your own latent: A new approach to self-supervised learning, 2020. URL https://arxiv.org/abs/2006.07733.

David Ha and Jürgen Schmidhuber. World models. *arXiv preprint arXiv:1803.10122*, 2(3):440, 2018.

Danijar Hafner, Timothy Lillicrap, Ian Fischer, Ruben Villegas, David Ha, Honglak Lee, and James Davidson. Learning latent dynamics for planning from pixels, 2019. URL https://arxiv.org/abs/1811.04551.

Lei Huang, Dawei Yang, Bo Lang, and Jia Deng. Decorrelated batch normalization, 2018. URL https://arxiv.org/abs/1804.08450.

Aapo Hyvarinen and Hiroshi Morioka. Unsupervised feature extraction by time-contrastive learning and nonlinear ica, 2016. URL https://arxiv.org/abs/1605.06336.

Aapo Hyvärinen and Hiroshi Morioka. Nonlinear ICA of temporally dependent stationary sources. In *Proceedings of the 20th International Conference on Artificial Intelligence and Statistics*, volume 54 of *Proceedings of Machine Learning Research*, pp. 460–469, 2017. URL https://proceedings.mlr.press/v54/hyvarinen17a.html.

Sergey Ioffe. Batch renormalization: Towards reducing minibatch dependence in batch-normalized models, 2017. URL https://arxiv.org/abs/1702.03275.

Sergey Ioffe and Christian Szegedy. Batch normalization: Accelerating deep network training by reducing internal covariate shift, 2015. URL https://arxiv.org/abs/1502.03167.

Li Jing, Pascal Vincent, Yann LeCun, and Yuandong Tian. Understanding dimensional collapse in contrastive self-supervised learning, 2022. URL https://arxiv.org/abs/2110.09348.

David Klindt, Yann LeCun, and Randall Balestriero. When does lejepa learn a world model?, 2026. URL https://arxiv.org/abs/2605.26379.

Chang Liu, Fei Suo, Yanzhou Jin, Zeyu Ping, Yusuke Iwasawa, Yutaka Matsuo, and Yaonan Zhu. Temporally centered sigreg improves leworldmodel representations for robot policy learning, 2026. URL https://arxiv.org/abs/2607.26924.

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

Hiroshi Morioka, Hermanni Hälvä, and Aapo Hyvärinen. Independent innovation analysis for nonlinear vector autoregressive process, 2021. URL https://arxiv.org/abs/2006.10944.

Aaron van den Oord, Yazhe Li, and Oriol Vinyals. Representation learning with contrastive predictive coding, 2019. URL https://arxiv.org/abs/1807.03748.

Laurenz Wiskott and Terrence J. Sejnowski. Slow feature analysis: Unsupervised learning of invariances. *Neural Computation*, 14(4):715–770, 2002. doi: 10.1162/089976602317318938.

Weiran Yao, Guangyi Chen, and Kun Zhang. Temporally disentangled representation learning, 2022a. URL https://arxiv.org/abs/2210.13647.

Weiran Yao, Yuewen Sun, Alex Ho, Changyin Sun, and Kun Zhang. Learning temporally causal latent processes from general temporal data, 2022b. URL https://arxiv.org/abs/2110.05428.

Jure Zbontar, Li Jing, Ishan Misra, Yann LeCun, and Stéphane Deny. Barlow twins: Self-supervised learning via redundancy reduction, 2021. URL https://arxiv.org/abs/2103.03230.

Kai Zhao, Dongliang Nie, Yuchen Lin, Zhehan Luo, Yixiao Gu, Deng-Ping Fan, and Dan Zeng. Sub-jepa: Subspace gaussian regularization for stable end-to-end world models, 2026. URL https://arxiv.org/abs/2605.09241.

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

A Mathematical motivation and scope

A.1 Oracle innovations and episodic pooling

We first ask what the residual would recover if the true latent state and transition coefficients were known; we call this the *oracle* construction. We then examine why estimating these quantities from embeddings need not recover the same noise. Fix a stride δ and let k index successive states sampled δ environment steps apart; here k indexes trajectory time, not training updates. Let $D = \mathrm{diag}(\rho_\delta)$ with $|\rho_{\delta,j}| < 1$, and assume

$$s_{k+1} = Ds_k + (I_n - D^2)^{1/2}\varepsilon_k, \qquad \varepsilon_k \overset{\text{iid}}{\sim} \mathcal{N}(0, I_n), \qquad \varepsilon_k \perp (s_0, \varepsilon_0, \ldots, \varepsilon_{k-1}). \tag{5}$$

The initial law of s_0 is arbitrary and independent of the noise sequence. This is a motivating model class, not an assumption verified for every benchmark. In particular, the implemented residual does not condition its coefficient on actions.

Lemma 1 (Oracle innovation under arbitrary initialization)**.** *Under Eq. equation 5, the oracle residual*

$$\eta_k^{\text{oracle}} = (I_n - D^2)^{-1/2}(s_{k+1} - Ds_k) \tag{6}$$

is standard Gaussian and independent of s_k for every k and every initial-state law. Nevertheless, the pooled state marginal over a finite episode can be non-Gaussian.

Proof. Substitution in Eq. equation 6 gives $\eta_k^{\text{oracle}} = \varepsilon_k$. Since s_k is a function of $s_0, \varepsilon_0, \ldots, \varepsilon_{k-1}$, the stated independence follows.

For a concrete non-Gaussian pooled marginal, take a scalar process with $0 < \rho < 1$ and $s_0 \sim \mathcal{N}(0, v_0)$, where $v_0 \neq 1$. Then $s_k \sim \mathcal{N}(0, v_k)$ with $v_k = 1 + \rho^{2k}(v_0 - 1)$. If K is uniform on $\{0, \ldots, H-1\}$, independent of the process, and $H \geq 2$, the pooled variable $S = s_K$ has

$$\mathbb{E}[S^2] = \overline{v}, \qquad \mathbb{E}[S^4] = 3\,\overline{v^2}, \qquad \frac{\mathbb{E}[S^4]}{(\mathbb{E}[S^2])^2} - 3 = \frac{3\,\mathrm{Var}_K(v_K)}{\overline{v}^{\,2}} > 0, \tag{7}$$

where bars average over K. Its excess kurtosis is positive, so its marginal is not Gaussian, although every oracle innovation is. □

The lemma is a possibility result. It does not assert that every nonstationary initialization produces a non-Gaussian pooled marginal, or that pooling preserves the modality of the initial distribution. The independent-noise construction uses a nonoverlapping stride grid. Oracle residuals formed from overlapping multi-step intervals can share noise; Gaussianity of each residual does not make the pooled residual samples independent.

A.2 Stationary equivalence and the scope of the motivation

Proposition 1 (Stationary Gaussian equivalence)**.** *Let $h_k \in \mathbb{R}^d$ be stationary and suppose*

$$h_{k+1} = Rh_k + (I_d - R^2)^{1/2}\xi_k, \qquad R = \mathrm{diag}(r_1, \ldots, r_d), \quad |r_j| < 1, \qquad \xi_k \perp h_k. \tag{8}$$

Then $h_k \sim \mathcal{N}(0, I_d)$ if and only if $\xi_k \sim \mathcal{N}(0, I_d)$.

Proof. First suppose $\xi_k \sim \mathcal{N}(0, I_d)$. Let $\varphi(u) = \mathbb{E}[\exp(\mathrm{i}u^\top h_k)]$ be the stationary characteristic function, where $\mathrm{i}^2 = -1$. Stationarity and independence imply

$$\varphi(u) = \varphi(Ru)\exp\big[-\tfrac{1}{2}u^\top(I_d - R^2)u\big] \tag{9}$$

$$= \varphi(R^m u)\exp\big[-\tfrac{1}{2}u^\top(I_d - R^{2m})u\big] \xrightarrow[m\to\infty]{} \exp(-\|u\|_2^2/2). \tag{10}$$

Here $R^m u \to 0$ because the coefficients are stable, and $\varphi(R^m u) \to \varphi(0) = 1$ by continuity. The limiting characteristic function determines the standard Gaussian law.

Conversely, suppose $h_k \sim \mathcal{N}(0, I_d)$, keeping the independence assumption but making no distributional assumption on ξ_k. With $Q = (I_d - R^2)^{1/2}$ and ψ the characteristic function of ξ_k, stationarity gives

$$e^{-\|u\|_2^2/2} = e^{-\|Ru\|_2^2/2}\psi(Qu), \qquad \psi(Qu) = e^{-\|Qu\|_2^2/2}. \tag{11}$$

The first factor never vanishes, and Q is invertible because $|r_j| < 1$. Thus $\psi(w) = e^{-\|w\|_2^2/2}$ for every $w \in \mathbb{R}^d$, proving $\xi_k \sim \mathcal{N}(0, I_d)$. □

The proposition identifies when the two Gaussian distributional targets are compatible within this model class. Equivalently, a non-Gaussian stationary state distribution rules out state-independent standard Gaussian innovations under the stated diagonal transition and noise scale. The episodic example in Lemma 1 avoids this restriction through nonstationarity. It does not justify Gaussian innovations for every stationary physical process. In particular, an exactly bounded, nondegenerate physical state is not Gaussian; a learned embedding need not retain its support or transition form. Performance on Two-Room is therefore an empirical finding, not a consequence of this proposition.

Equivalence of these distributional conditions does not make finite regularized objectives or their optimization trajectories equivalent. The proposition also does not establish identifiability of a hidden state. Applying an identification theorem requires its assumptions on the true transition, observation map, representation dimension, objective, and optimizer. For example, the isotropic stationary setting analyzed by Klindt et al. (2026) is more restrictive than arbitrary diagonal dynamics with a learned predictor. We do not extend its alignment equality case to our episodic, finite-penalty objective. Neither independence of residual and state nor exact population Gaussianity is imposed by a finite SIGReg loss.

A.3 Why the estimated residual need not be an oracle innovation

Even an identity encoder, whose embedding equals the true state, need not recover the oracle innovation after standardization. Consider one coordinate with source state s, next state s', and true dynamics $s' = \rho s + \sqrt{1-\rho^2}\varepsilon$, where $|\rho| < 1$ and $\varepsilon \sim \mathcal{N}(0,1)$ is independent of s. Write μ, $v > 0$, and r for the common normalization mean, variance, and selected persistence estimate in Eq. equation 3. Unlike the true coefficient ρ, r is supplied by the estimator. For $|r| < 1$ and an inactive variance floor, substitution gives

$$\eta = \frac{(\rho - r)s + (r-1)\mu + \sqrt{1-\rho^2}\,\varepsilon}{\sqrt{v(1-r^2)}}. \tag{12}$$

Thus a coefficient error leaves a state-dependent term. With fixed normalizers and an exact, unclipped coefficient $r = \rho$, the result is $\eta = \varepsilon/\sqrt{v} + (\rho-1)\mu/\sqrt{v(1-\rho^2)}$; arbitrary transient means and variances do not generally yield a standard Gaussian residual. In this exact-coefficient case the residual remains Gaussian, but its mean and variance can be wrong. For source-only population normalization, the fitted slope equals ρ when the source has finite, positive variance and the noise is independent of it (Appendix B.1). Consequently,

$$\mathbb{E}[\eta] = \frac{(\rho-1)\mu}{\sqrt{v(1-\rho^2)}}, \qquad \mathrm{Var}(\eta) = \frac{1}{v}, \tag{13}$$

where μ and v are source moments in the stationary units of Eq. equation 5. The oracle noise is standard Gaussian; the empirically normalized target need not be.

This distinction matters because affine reparameterization cannot repair the mismatch when batch moments are recomputed: positive coordinatewise rescaling and translation leave η unchanged

(Appendix B.5). For the exact-coefficient, population construction, subtraction removes non-Gaussian shape inherited from the transient source but can leave location and scale errors. That conclusion does not extend to an erroneous fitted coefficient: all-frame pooling, finite batches, EMA lag, or clipping can leave the state-dependent term in Eq. equation 12. With data-dependent estimates, even the finite-batch residual distribution need not be exactly Gaussian.

The diagonal construction also depends on the learned coordinate system. For an invertible mixing matrix A, the embedding $e = As$ has transition ADA^{-1}, which need not be diagonal when D has unequal entries. Coordinatewise standardization does not remove that coupling. These facts motivate evaluating the practical residual on decoding and planning tasks, without interpreting a low residual regularization loss as a certificate of state recovery.

B Method and implementation details

B.1 Normalization samples, temporal pairs, and batch statistics

Let E be the raw embeddings in a training batch, $\mathcal{P}$ the set of valid within-trajectory index pairs (i, i'), and $\mathcal{S}$ the indices used to estimate the normalization statistics. Write $M = |\mathcal{S}|$ and $N = |\mathcal{P}|$. Thus M counts embeddings used for normalization and N counts transitions used to estimate persistence; these sample sets need not coincide. Bars denote current-batch estimates, and $\widetilde{z}$ uses these batch moments. The final z in Eq. equation 2 instead uses the selected starred statistics, which may be running averages. The coordinatewise batch statistics are

$$\overline{\mu}_j = \frac{1}{M}\sum_{i\in\mathcal{S}} e_{i,j}, \qquad \overline{v}_j = \max\left\{\frac{1}{M}\sum_{i\in\mathcal{S}}(e_{i,j}-\overline{\mu}_j)^2, \epsilon_v\right\}, \tag{14}$$

$$\widetilde{z}_{i,j} = \frac{e_{i,j}-\overline{\mu}_j}{\sqrt{\overline{v}_j}}, \qquad \overline{\rho}_j = \frac{1}{N}\sum_{(i,i')\in\mathcal{P}} \widetilde{z}_{i,j}\widetilde{z}_{i',j}. \tag{15}$$

The variance uses divisor M, corresponding to `unbiased=False`. The same statistics standardize both endpoints. The lag product does not separately recenter the next endpoint or divide by its own standard deviation, so its interpretation depends on the normalization sample set.

For LeWM, a batch contains B sequences of T sampled frames, giving $E \in \mathbb{R}^{B\times T\times d}$ ($T = 4$ in the reported setup). The normalization set $\mathcal{S}$ includes all BT frames and $\mathcal{P} = \{((b,t),(b,t+1)) : 1 \le b \le B, 1 \le t < T\}$. Each frame enters the mean and variance once, including interior frames that participate in two lag pairs. For the pairwise Reacher form, $E^-, E^+ \in \mathbb{R}^{B\times d}$ contain endpoints separated by δ, $\mathcal{P} = \{(i^-, i^+) : 1 \le i \le B\}$, and normalization uses the source endpoints $\mathcal{S} = \{i^-\}_{i=1}^{B}$. Temporal pairs never cross an episode boundary. Sequence-adjacent pairing is lag one in the supplied sequence; any environment-frame subsampling belongs to the data configuration.

What the fitted coefficient estimates. In Reacher, source centering gives $\sum_i (e^-_{i,j} - \overline{\mu}_j) = 0$. With an inactive variance floor,

$$\overline{\rho}_j = \frac{\widehat{\mathrm{Cov}}(e^-_j, e^+_j)}{\widehat{\mathrm{Var}}(e^-_j)}. \tag{16}$$

Both sample moments use divisor B. This is exactly the least-squares slope with an intercept, even if target and source moments differ; it is not generally their Pearson correlation. The residual itself uses the constrained intercept $(1-\overline{\rho}_j)\overline{\mu}_j$, rather than the regression intercept $\overline{\mu}^+_j - \overline{\rho}_j\overline{\mu}_j$. Under Eq. equation 5, the population slope is $\rho_{\delta,j}$ whenever the source variance is finite and positive. The sample slope is consistent when the empirical moments converge, for example across independent episodes of fixed finite length with finite second moments. The oracle identity in Lemma 1 needs no such moment assumption; statistical consistency does.

In LeWM, normalization includes all frames, while regression sources exclude each sequence's last frame and targets exclude its first. The resulting lag product generally differs from both a regression slope and a correlation coefficient. It equals both when the source and target means and variances agree with the pooled moments. Thus all-frame pooling and source-only normalization must not be interchanged in either the implementation or its interpretation.

Table 3: Statistics used by the residual. Every mode preserves the direct gradient from η to the encoder outputs. These are three evaluated choices, forming two one-factor contrasts, not a complete factorial design.

Mode	Selected $(\mu^*, v^*, \widehat{\rho}^*)$	Statistics gradients
EMA-detached	Updated EMA buffers m_k	Blocked
Batch-detached	$\mathrm{sg}(\overline{\mu}, \overline{v}, \overline{\rho})$	Blocked
Batch-differentiable	$(\overline{\mu}, \overline{v}, \overline{\rho})$	Retained

Which moments the residual actually matches. For one coordinate, let m_-, m_+, v_-, v_+, and c_{-+} denote the source and target means, variances, and covariance under the pair distribution. Given fixed common normalizers (μ, v) and $|r| < 1$,

$$\mathbb{E}[\eta] = \frac{m_+ - r m_- - (1-r)\mu}{\sqrt{v(1-r^2)}},$$
$$\mathrm{Var}(\eta) = \frac{v_+ + r^2 v_- - 2r c_{-+}}{v(1-r^2)}. \tag{17}$$

The same algebra holds exactly for empirical moments over a batch when the common normalizers and coefficient take their fitted values. For the source-only batch layout, with the floor inactive and $\overline{\rho}_j$ unclipped, this reduces to

$$\overline{\eta}_j = \frac{\overline{\mu}_j^+ - \overline{\mu}_j}{\sqrt{\overline{v}_j(1-\overline{\rho}_j^2)}}, \qquad \widehat{\mathrm{Var}}(\eta_j) = \frac{\overline{v}_j^+ / \overline{v}_j - \overline{\rho}_j^2}{1-\overline{\rho}_j^2}. \tag{18}$$

Here $\overline{\mu}_j^+$ and $\overline{v}_j^+$ are the target batch moments and $\overline{\eta}_j$ is the residual batch mean. In this layout the residual has zero mean and unit variance exactly when the two endpoint means and variances agree. Its empirical covariance with the source is zero, but this regression property does not establish independence. Once clipping or a variance floor is active, the simplified identities need not hold; Eq. equation 17 still describes the fixed-coefficient residual with the effective normalization variance.

B.2 EMA UPDATES AND THE THREE STATISTICS MODES

The running state consists of $m_k = (\mu_k, v_k, \widehat{\rho}_k)$, indexed by training update k. On the first training batch the buffers copy the batch statistics. Subsequent updates use

$$m_k = (1-\beta) m_{k-1} + \beta\ \mathrm{sg}(\overline{\mu}_k, \overline{v}_k, \overline{\rho}_k), \qquad \beta = 0.01. \tag{19}$$

The operator sg acts componentwise on a tuple: its forward value is unchanged and its derivative is zero. EMA updates are performed without gradient tracking. The batch lag product entering the EMA is computed using *batch* moments, as in Eq. equation 15, not the running moments. The updated buffers are available to that step's residual. Current-batch modes retain this detached running-statistics stream, but use the live batch statistics in the loss.

The selected lag estimate is clipped to $[-0.995, 0.995]$ after statistics selection. The embedding variance is floored at 10^{-8}; there is no additional additive epsilon in the diagonal residual denominator. Clipping ensures $1 - (\rho_j^*)^2 \geq 1 - 0.995^2 = 0.009975$, limiting that residual-rescaling factor to approximately 10.01. This does not bound the full gradient, which also depends on embedding variance and the statistics path. Within a differentiable mode, the clamp and variance floor retain their usual piecewise derivatives.

EMA-detached versus batch-detached changes the age of the statistics while keeping their derivatives blocked. Batch-detached versus batch-differentiable keeps the forward estimator fixed and changes its derivative. The matched three-mode study is on Reacher. On Two-Room, the first prospective cohort's innovation, fixed-anchor, scheduled-anchor, guarded, normalization, and differencing arms all use differentiable batch statistics; the second cohort compares the two batch modes. The earlier Two-Room EMA results come from a separate historical cohort and do not constitute a matched freshness contrast.

B.3 SIGREG AND THE COMPLETE LOSS

For a sample matrix $Y = (y_1, \dots, y_N)^\top \in \mathbb{R}^{N \times d}$, let $a_1, \dots, a_K$ be random unit projection directions. Each projection produces scalar samples $a_k^\top y_i$. A characteristic function describes a distribution through expected complex exponentials; its empirical version replaces the expectation with a sample average. SIGReg compares this function with the standard Gaussian characteristic function $\exp(-u^2/2)$ using the Epps–Pulley discrepancy (Epps & Pulley, 1983; Balestriero & LeCun, 2025). Numerically integrating the squared discrepancy gives

$$\widehat{\varphi}_k(u) = \frac{1}{N} \sum_{i=1}^{N} \exp(\mathrm{i} u a_k^\top y_i), \tag{20}$$

$$\mathcal{G}(Y) = \frac{N}{K} \sum_{k=1}^{K} \sum_{\ell=1}^{L} q_\ell \left| \widehat{\varphi}_k(u_\ell) - \exp(-u_\ell^2/2) \right|^2 . \tag{21}$$

Here u_ℓ are the L frequency points at which the characteristic functions are compared, not temporal lags. The weights q_ℓ combine the trapezoidal numerical-integration rule with a Gaussian weighting window. In this subsection, N counts samples in one SIGReg evaluation; it need not equal the earlier count of transition pairs. The imaginary unit is i; the modulus squared sums the squared real and imaginary discrepancies. The factor N is part of the statistic, so changing sample aggregation changes its scale. We retain the native SIGReg implementations, using $K = 256$ projections for Reacher and $K = 1024$ for LeWM. With $L = 17$ knots, $u_\ell = (\ell - 1)\Delta$ and $\Delta = 3/(L-1)$, their quadrature weights are

$$q_\ell = \exp(-u_\ell^2/2) \begin{cases} \Delta, & \ell \in \{1, L\}, \\ 2\Delta, & 1 < \ell < L. \end{cases} \tag{22}$$

The doubled half-axis weights account for the even integrand on $[-3, 3]$. Projection directions are normalized Gaussian vectors, resampled at each call and shared across that call's time positions.

For $H \in \mathbb{R}^{V \times B \times d}$, V indexes endpoint views or sampled time positions, B is batch size, and d is embedding dimension. The pipelines aggregate these samples differently:

$$\mathrm{SIGReg}_{\mathrm{R}}(H) = \mathcal{G}(\mathrm{flatten}_{V,B}(H)), \tag{23}$$

$$\mathrm{SIGReg}_{\mathrm{L}}(H) = \frac{1}{V} \sum_{v=1}^{V} \mathcal{G}(H_{v,:,:}). \tag{24}$$

Reacher pools the endpoint views for the raw anchor ($V = 2$, $N = 2B$) and uses one residual per pair ($V = 1$, $N = B$). LeWM applies the statistic across the batch separately at each time position and then averages over time, following its stepwise regularization (Maes et al., 2026). Its inputs swap the first two axes of the raw $B \times T \times d$ tensor and residual $B \times (T-1) \times d$ tensor, so $V = T$ and $V = T - 1$, respectively, with $N = B$ in each evaluation of $\mathcal{G}$. These reductions and sample-count factors are part of the loss convention.

To see the role of sample count, consider independent samples and fix a projection and frequency. If φ is the corresponding population characteristic function and $\varphi_0(u) = e^{-u^2/2}$, then

$$\mathbb{E}\big[N|\widehat{\varphi}(u) - \varphi_0(u)|^2\big] = N|\varphi(u) - \varphi_0(u)|^2 + 1 - |\varphi(u)|^2. \tag{25}$$

Thus N multiplies the population discrepancy, while the iid sampling term stays bounded. The Reacher anchor and residual use different N, whereas each LeWM evaluation uses $N = B$ for both. Neither λ nor w_{m} has a common numerical interpretation across the two pipelines. Equation equation 25 explains the iid scaling; temporal dependence can alter its sampling term in actual trajectory batches.

We preserve the loss coefficients used by the corresponding baseline:

$$\mathcal{L}_{\mathrm{LeWM}} = \mathcal{L}_{\mathrm{pred}} + \lambda_{\mathrm{L}} \big[\mathrm{SIGReg}(\eta) + w_{\mathrm{m}}\, \mathrm{SIGReg}(e) \big], \tag{26}$$

$$\mathcal{L}_{\mathrm{Reacher}} = (1 - \lambda_{\mathrm{R}}) \mathcal{L}_{\mathrm{pred}} + \lambda_{\mathrm{R}} \big[\mathrm{SIGReg}(\eta) + w_{\mathrm{m}}\, \mathrm{SIGReg}(e) \big]. \tag{27}$$

Let $\widehat{E}$ be the LeWM predictor's output given its embedding context and encoded actions, and E^{tgt} the corresponding future embeddings. For their common shape $B \times T_{\mathrm{pred}} \times d$, the prediction term

is $\mathcal{L}_{\mathrm{pred}} = \|\widehat{E} - E^{\mathrm{tgt}}\|_F^2/(BT_{\mathrm{pred}}d)$, where T_{pred} counts target positions. The Reacher alignment term is the mean-squared deviation of the two endpoint embeddings from their view mean; for two views it equals $\mathcal{L}_{\mathrm{pred}} = \|E^+ - E^-\|_F^2/(4Bd)$. The symbol $\mathcal{L}_{\mathrm{pred}}$ denotes this inherited term in each pipeline; Reacher's version does not introduce a learned predictor.

The prospective Two-Room innovation configurations use $\lambda_{\mathrm{L}} = 0.09$ and $w_{\mathrm{m}} = 0.1$ unless an anchor intervention is specified. The prospective stride-64 Reacher innovation configuration uses $\lambda_{\mathrm{R}} = 0.005$ and $w_{\mathrm{m}} = 0.1$; its marginal comparator uses 0.05. These are previously selected, benchmark-specific settings. The stride-16 innovation configuration instead uses $\lambda_{\mathrm{R}} = 0.01$ and $w_{\mathrm{m}} = 1$, and should not be described as using a uniformly small anchor.

Finite sketches encourage agreement with a Gaussian target but do not certify joint Gaussianity, conditional Gaussianity, or temporal independence. Correlation among within-trajectory samples also prevents interpreting the training statistic as a calibrated iid goodness-of-fit test without a dependence-aware reference distribution.

B.4 Training step and gradient paths

Algorithm 1 separates the detached running-statistics stream from the statistics actually used by the residual. The function $\mathrm{BatchStats}$ denotes Eqs. equation 14–equation 15. The selected statistics must be available before the final standardization of the live embeddings.

Algorithm 1 Innovation SIGReg, one training update (diagonal form)

Require: Trajectories; statistics mode; buffers m; λ, w_{m}, α_λ

Compute raw regularizer embeddings E and the original loss $\mathcal{L}_{\mathrm{pred}}$
Form normalization indices $\mathcal{S}$ and within-trajectory pairs $\mathcal{P}$
$\overline{m}^{\log} \leftarrow \mathrm{BatchStats}(\mathrm{sg}(E), \mathcal{S}, \mathcal{P})$
Initialize or update m using $\overline{m}^{\log}$ and Eq. equation 19
$w \leftarrow w_{\mathrm{m}}$; if enabled, update the detached guard and w (Section B.6)
if mode is EMA-detached **then**
    $(\mu^*, v^*, \widehat{\rho}^*) \leftarrow m$ ▷ no statistics gradients
else
    $(\mu^*, v^*, \widehat{\rho}^*) \leftarrow \mathrm{BatchStats}(E, \mathcal{S}, \mathcal{P})$
    if mode is batch-detached **then**
        $(\mu^*, v^*, \widehat{\rho}^*) \leftarrow \mathrm{sg}(\mu^*, v^*, \widehat{\rho}^*)$
    end if
end if
$\rho^* \leftarrow \mathrm{clip}(\widehat{\rho}^*, -\rho_{\max}, \rho_{\max})$
$Z \leftarrow (E - \mu^*) \oslash \sqrt{\max(v^*, \epsilon_v)}$ ▷ recompute from live E
$\eta_{ii'} \leftarrow (Z_{i'} - \rho^* \odot Z_i) \oslash \sqrt{\mathbf{1} - (\rho^*)^2}$ for $(i, i') \in \mathcal{P}$
$\mathcal{L} \leftarrow \alpha_\lambda \mathcal{L}_{\mathrm{pred}} + \lambda[\mathrm{SIGReg}(\eta) + w\,\mathrm{SIGReg}(E)]$
Backpropagate $\mathcal{L}$ and update the model using the original optimizer

For detached statistics, and considering a single pair with the statistics held fixed, the direct Jacobians are

$$\frac{\partial \eta}{\partial e^+} = \mathrm{diag}\Big([v^* \odot (\mathbf{1} - (\rho^*)^2)]^{-1/2}\Big), \qquad \frac{\partial \eta}{\partial e^-} = -\,\mathrm{diag}\Big(\rho^* \odot [v^* \odot (\mathbf{1} - (\rho^*)^2)]^{-1/2}\Big), \tag{28}$$

where v^* in this expression includes the numerical floor. These Jacobians demonstrate the remaining paths to the encoder. Differentiable batch statistics add derivatives through the mean, variance, and lag estimate, coupling examples in the batch. Detaching Z itself would remove the direct paths in Eq. equation 28 and would implement a different intervention.

A matched forward comparison must reuse the input embeddings, model state, and random projection draws for each SIGReg call. Otherwise sketch randomness can change the loss even before a model update. Identical forward values at this fixed state do not imply identical training trajectories after the gradients are applied. The experiment also differs from stopping a target encoder or maintaining an EMA of encoder parameters; the running averages here contain statistics only.

B.5 AFFINE INVARIANCE AND ITS GRADIENT CONSEQUENCES

The residual uses normalization to remove coordinatewise location and scale. This gives an exact invariance of the current-batch computation whenever the variance floor is inactive. For constants $a_j > 0$ and b_j, transform every embedding coordinate by $e'_{i,j} = a_j e_{i,j} + b_j$. Recomputing the statistics gives

$$\overline{\mu}'_j = a_j \overline{\mu}_j + b_j, \qquad \overline{v}'_j = a_j^2 \overline{v}_j, \qquad z'_{i,j} = z_{i,j}, \qquad \overline{\rho}'_j = \overline{\rho}_j, \qquad \eta'_{ii',j} = \eta_{ii',j}. \tag{29}$$

Both the original and transformed variances must exceed the floor. Clipping does not change the invariance because its input is unchanged. For EMA normalization it also holds if the full buffer history is transformed consistently; after changing embeddings with fixed buffers, it generally holds only as the buffers adapt.

Fix the random sketches and let $\mathcal{L}_\eta(E) = \mathrm{SIGReg}(\eta(E))$ denote only the residual penalty with differentiable batch statistics. Differentiating the invariance with respect to b_j and a_j at $(a_j, b_j) = (1, 0)$ yields

$$\sum_i \frac{\partial \mathcal{L}_\eta}{\partial e_{i,j}} = 0, \qquad \sum_i e_{i,j} \frac{\partial \mathcal{L}_\eta}{\partial e_{i,j}} = 0. \tag{30}$$

The sums include every embedding row in the batch, including both endpoints. At differentiable points these identities make the embedding gradient orthogonal to coordinatewise translations and rescalings. They apply to the residual term, not to the raw anchor or the complete training loss.

With detached statistics, backpropagation instead takes a partial derivative holding the normalizers and coefficient fixed. This derivative need not satisfy Eq. equation 30. A change consisting solely of a common coordinatewise shift or positive rescaling is removed once the batch statistics are recomputed; a general optimizer step also changes embedding shape, so its total effect need not vanish. The cancellation of a bias update by subsequent centering is a related motivation for differentiating through batch-normalization statistics (Ioffe & Szegedy, 2015). These identities explain why the backward paths differ, but do not predict which yields better representations: the encoder parameterization, other loss terms, finite steps, and EMA lag also affect training.

B.6 RAW-EMBEDDING ANCHOR AND OPTIONAL GUARD

The invariance in Eq. equation 29 makes a raw anchor useful even when residual matching succeeds. Under uniform rescaling $e \mapsto \gamma e$, Reacher's alignment loss scales by γ^2, while the batch residual penalty is unchanged as long as the floor stays inactive. LeWM's prediction loss likewise scales by γ^2 if its prediction and target are both rescaled; whether a given predictor realizes that transformation depends on its architecture. Thus the residual term alone need not resist scale reduction.

The raw anchor compares unnormalized embeddings with a fixed standard Gaussian target and retains pressure on their location and scale. It is a soft penalty on the same marginal used by the baseline, so it does not remove the potential mismatch with non-Gaussian state laws. Scale sensitivity alone does not prove that all collapsed stationary points are absent. In particular, the invariance argument stops applying when shrinking activates the numerical floor.

For the optional guard, flatten the raw LeWM embeddings into $E \in \mathbb{R}^{M \times d}$, subtract the column means across samples, and form the covariance C_k. Let $\ell_{k,1}, \ldots, \ell_{k,d}$ be its nonnegative eigenvalues. The participation ratio is

$$r_k = \frac{(\sum_j \ell_{k,j})^2}{\sum_j \ell_{k,j}^2}, \qquad \overline{r}_k = 0.95\, \overline{r}_{k-1} + 0.05\, r_k, \tag{31}$$

for nonzero covariance. The ratio measures how many variance-carrying directions contribute appreciably; it is not the algebraic rank. The monitor is computed without gradients, in float32 with autocasting disabled. With latch $g_k \in \{0, 1\}$ initially inactive, the intervention is

$$g_k = g_{k-1} \vee [\overline{r}_k < r_{\min}], \qquad w_k = \begin{cases} w_{\mathrm{boost}}, & g_k = 1, \\ w_{\mathrm{m}}, & g_k = 0. \end{cases} \tag{32}$$

The updated w_k is used in the current loss and remains boosted after the first trigger. Two-Room uses $r_{\min} = 12$, $w_{\mathrm{m}} = 0.1$, and $w_{\mathrm{boost}} = 0.5$. These thresholds are specific to this online monitor. They

should not be applied to standardized embeddings or to covariance estimates from a different offline sampling protocol without recalibration.

The fixed-anchor control uses $w_k = 0.5$ throughout. The scheduled control switches from 0.1 to 0.5 at update 4960, corresponding to 8% of the planned 62000 updates, independently of the monitor. Both retain the differentiable batch residual used by the guarded arm. The guard is an intervention on regularization strength, not a validated classifier of future planning failures. Its threshold can be crossed by runs that would otherwise finish successfully.

All eight guarded models in the two planning cohorts triggered between updates 3,978 and 5,469 (Table 23); in the first cohort, four triggers precede the scheduled switch and one follows it. These timings do not establish that an adaptive trigger is necessary for good performance.

B.7 Controls and supplementary variants

The marginal baseline applies SIGReg to raw embeddings, with no residual term or additional anchor. To place the normalization and differencing controls in a common family, write

$$\eta(r, \sigma) = (z^+ - r \odot z^-) \oslash \sigma, \qquad \sigma_j > 0. \tag{33}$$

Normalization-only uses $r = \mathbf{0}$ and $\sigma = \mathbf{1}$, and thus regularizes the standardized next embedding. It is the $\rho^* = 0$ case of Eq. equation 3 when the normalizer is matched. Temporal differencing uses $r = \mathbf{1}$ and an estimated coordinatewise difference scale σ^Δ. It is not a continuous extension of the AR normalization to $\rho^* = 1$, where that denominator vanishes. Innovation SIGReg instead uses $r = \rho^*$ and $\sigma = \sqrt{\mathbf{1} - (\rho^*)^2}$. These comparisons change subtraction and scaling together, so they do not separately identify the contribution of either operation.

Both simple controls keep the raw anchor with $w_{\mathrm{m}} = 0.1$. The Two-Room normalization control in the first cohort and differencing controls in both cohorts select differentiable batch statistics through their launch configuration. Reacher's prospective stride-64 comparisons use the EMA-detached configuration.

The shared mean cancels from the difference numerator. A stronger cancellation holds if the scale is computed as the current-batch standard deviation of that same standardized difference. In one coordinate, let $\Delta e = e^+ - e^-$ and use any common positive normalization variance v^*. With an inactive difference-variance floor,

$$\frac{\Delta e / \sqrt{v^*}}{\sqrt{\widehat{\mathrm{Var}}(\Delta e / \sqrt{v^*})}} = \frac{\Delta e}{\sqrt{\widehat{\mathrm{Var}}(\Delta e)}}. \tag{34}$$

This identity requires the scale to be recomputed from the current batch; an EMA of difference variances at earlier embedding scales need not obey it, and detaching the scale can change the backward path even when the forward identity holds.

A supplementary full-matrix construction estimates a transition on standardized coordinates and whitens the corresponding residual with its estimated covariance. With C_{z^+z} and C_{zz} denoting cross- and source-covariance estimates, respectively, it takes the form

$$\widehat{R} = C_{z^+z}(C_{zz} + \epsilon_R I_d)^{-1}, \qquad u = z^+ - \widehat{R} z, \qquad \eta = \widehat{C}_u^{-1/2} u, \tag{35}$$

where $\epsilon_R > 0$ regularizes the covariance inverse and the eigenvalues of $\widehat{C}_u$ are floored at $1 - 0.995^2$ before the inverse square root. This supplementary variant estimates cross-coordinate dynamics and is separate from the diagonal configurations above.

B.8 State, inference, and computational cost

The diagonal statistics require $O(Md + Nd)$ arithmetic and $O(d)$ running state; materialized embeddings and residuals use $O(Md + Nd)$ storage. These costs exclude SIGReg's projection/quadrature evaluations and the optional covariance-rank monitor. The raw anchor adds a SIGReg evaluation.

Resuming training requires the normalization buffers, initialization flag, guard state, and optimizer state. These statistics are module buffers in full training checkpoints, but may be absent from model-only exports. Planning uses the learned encoder and predictor without evaluating the regularizer.

C DATASETS, EVALUATION, AND COHORT PROTOCOLS

C.1 TRAINING CONFIGURATIONS AND EVALUATION TARGETS

The Reacher experiments use random-policy trajectories generated with the upstream collection recipe, with 10,000 episodes and 100,000 sampled training pairs per temporal stride. A CNN maps 64×64 images to two-dimensional embeddings. Training uses 100 epochs, batch size 256, AdamW with learning rate 3×10^{-3}, and 256 SIGReg projections. The loss follows the Reacher convention $(1-\lambda)\mathcal{L}_{\text{pred}} + \lambda[\text{SIGReg}(\eta) + w_{\text{m}}\,\text{SIGReg}(e)]$. Here e is the raw embedding and η the estimated transition residual defined in the method; the marginal baseline uses only SIGReg on e. The main stride-64 residual-family comparisons use $\lambda = 0.005$, $w_{\text{m}} = 0.1$, and EMA-detached statistics, whereas the marginal baseline uses $\lambda = 0.05$. The stride-16 innovation setting uses $\lambda = 0.01$ and $w_{\text{m}} = 1$. These settings were selected from development experiments separately for each method and then fixed for the reported fresh cohort. The comparison therefore assesses selected configurations, rather than changing the regularizer at one common regularization coefficient.

The development search was unequal across methods. At stride 64, the standard marginal scan used $\lambda \in \{0.001, 0.005, 0.01, 0.05\}$; additional marginal cells used 0.02 and 0.1. Innovation used the four standard weights with anchor weights 0.1 and 1, followed by additional anchor, statistics-mode, and duration variants. Table 9 reports all cells, including full-matrix and coefficient-clamp experiments at other strides. Normalization-only and temporal-difference controls were not tuned separately. These comparisons therefore evaluate the selected recipes and their controls, not equal-budget optimization of every regularizer.

The ten-run historical marginal cell at $\lambda = 0.1$ has mean $R^2 = 0.547 \pm 0.101$, close to the selected $\lambda = 0.05$ cell's 0.545 ± 0.071. The fresh comparison does not evaluate $\lambda = 0.1$, so its 0.251 gain is relative to the frozen 0.05 baseline, not a claim about every tuned marginal configuration. Historical and fresh scores are not combined to create an unperformed comparison.

The Reacher linear probe is fitted with an intercept, from learned embeddings to the two raw joint angles in radians. Its fitting observations come from trajectories, while its evaluation observations are 10,000 independent configurations with standard-normal angles, rendered using a fixed evaluation seed. For coordinate $j \in \{1, 2\}$, let y_{ij} and $\widehat{y}_{ij}$ be the true and predicted angle of test observation i, and let $\overline{y}_j$ be the test-set mean. We compute

$$R_j^2 = 1 - \frac{\sum_{i=1}^{N}(y_{ij} - \widehat{y}_{ij})^2}{\sum_{i=1}^{N}(y_{ij} - \overline{y}_j)^2}, \qquad R^2 = \frac{R_1^2 + R_2^2}{2}, \qquad R_{\min}^2 = \min(R_1^2, R_2^2). \tag{36}$$

N is the number of evaluation observations. Negative scores are retained. The primary summaries average R^2 and $R_{\min}^2$ separately across trained models. In particular, mean $R_{\min}^2$ is not obtained by taking the minimum of the two across-model coordinate means. We retain the evaluation of Klindt et al. (2026), which scores on independently rendered Gaussian-angle configurations; our written protocol had specified episode-disjoint trajectory evaluation, so results apply to the executed task.

Two-Room uses the LeWM architecture with a ViT-tiny encoder trained from scratch, hidden dimension 192, 12 encoder blocks, three encoder attention heads, patch size 14, and 224×224 input images. The predictor is a six-block transformer with 192-dimensional hidden states, 16 attention heads, MLP width 2,048, dropout 0.1, and a three-frame context. The projector is a $192 \to 2048 \to 192$ MLP with batch normalization. The complete module has approximately 18.0 million trainable parameters. Training uses 100 epochs, batch size 128, AdamW with learning rate 5×10^{-5}, and bfloat16 mixed precision. The available training sequence has four frames, sampled with frameskip 5. The loss uses the LeWM convention $\mathcal{L}_{\text{pred}} + 0.09\mathcal{L}_{\text{reg}}$. The 0.1 appearing in several configurations is the raw-anchor weight, not the outer regularization coefficient.

The main Two-Room dataset was regenerated using the upstream collector. It contains 2,000 episodes and 122,894 steps, with mean episode length 61.4. The official dataset contains 10,000 episodes and 920,809 steps, with mean length 92.1. Thus these datasets differ in both sample count and trajectory-length distribution. Their comparison does not isolate data size or a change in expert behavior. An official-data 100-epoch run took about 30 GPU-hours, and a regenerated-data profiling run took 2.75 GPU-hours. These are configuration-specific compute observations from different profiles, not a controlled efficiency comparison between regularizers.

All models within a planning cohort are evaluated with the same planner configuration, evaluation seeds $\{0, 1, 2, 42\}$, and 50 episodes per seed. Two-Room evaluation permits at most 100 environment steps per episode. For trained model r, let $b_{rkj} \in \{0, 1\}$ indicate success on episode j generated under evaluation seed k. Its score is

$$\mathrm{SR}_r = \frac{100}{4} \sum_{k \in \{0,1,2,42\}} \frac{1}{50} \sum_{j=1}^{50} b_{rkj}. \tag{37}$$

The reported mean and sample SD use these model-level scores as their inputs. Four evaluations of one model do not constitute four independently trained models. Evaluation seeds also select environment cases, so their dispersion includes case-set differences.

C.2 Comparison arms and cohort construction

Table 4 maps the compact arm labels to the tested configurations. These labels index the experiment records; they are not additional method names. The Reacher label N and Two-Room label Z denote the same type of normalization-only control in their respective codebases. The Two-Room I and J labels denote the same batch-differentiable innovation configuration in different cohorts.

Table 4: **Two-Room arm definitions.** The outer weight is 0.09 in every row. Innovation-family anchors act on raw embeddings. The fixed, scheduled, and guarded arms retain batch-differentiable residual statistics.

Arm	Regularization configuration
M	Marginal SIGReg on raw embeddings, without a separate residual or anchor.
I / J	Innovation SIGReg with current-batch differentiable mean, variance, and lag estimate; $w_{\mathrm{m}} = 0.1$.
B	Same current-batch innovation computation as J, with statistics detached and the standardized embeddings recomputed from live encoder outputs.
H	I with fixed $w_{\mathrm{m}} = 0.5$.
S	I with w_{m} increased from 0.1 to 0.5 at update 4,960 of 62,000 planned updates, without consulting a PR measurement.
G	I with a latched increase from 0.1 to 0.5 when the online smoothed participation ratio falls below 12.
Z	SIGReg on the standardized next embedding, without subtracting an estimated transition; $w_{\mathrm{m}} = 0.1$.
T	SIGReg on the standardized difference $z_{t+\delta} - z_t$, divided by an estimated coordinatewise difference scale; $w_{\mathrm{m}} = 0.1$.
U	Sub-JEPA regularizer with 32 frozen random orthonormal subspaces and the subspace-averaged Epps–Pulley statistic; no innovation anchor.
C	TC-LeWM adaptation: subtract the four-frame within-sequence mean of raw embeddings and apply SIGReg to the centered values; no autoregressive rescaling, standardization, or marginal anchor.

The first planning protocol was recorded before the new runs were launched. It contains seven core arms at nominal training seeds 1001–1005. Temporal differencing was an optional three-run arm, selected before fresh outcomes using a criterion on historical embedding autocorrelations. It uses labels 1001–1003; the eight empty evaluation rows for labels 1004–1005 in the CSV are unexecuted placeholders, not failed runs. This yields 38 trained models and 152 completed evaluations. The corresponding Reacher cohort contains 40 stride-64 runs (M/I/N/T at seeds 2001–2010) and ten stride-16 runs (M/I at seeds 2001–2005). Development smoke runs were not part of either fresh cohort.

The second planning cohort contains M/G/T/C/J/B at new nominal labels 4101–4103, totaling 18 models and 72 evaluations. Its choices were informed by the first cohort and specified before its own outcomes were inspected. It is a separate cohort, not a prespecified extension of the first. The Reacher statistics-mode comparison uses labels 5201–5203 for each of EMA-detached, batch-detached, and batch-differentiable statistics, at $\delta = 64$, $\lambda = 0.005$, and $w_{\mathrm{m}} = 0.1$.

The planned first-cohort primary planning contrasts were G–M, G–U, and G–S; secondary contrasts included I–Z, G–H, and I–M. The second-cohort contrasts were T–G, G–C, T–C, and B–J. A three-percentage-point band was used descriptively. It is not an equivalence test, a confidence interval, or

a statistical-significance threshold. All-run means are primary; thresholded outcomes and success-conditioned summaries are secondary. We do not infer a zero future failure probability from a small cohort without observed failures.

For Reacher, mean R^2 and mean $R^2_{\min}$ at stride 64 were co-primary outcomes, with I–M, I–N, and I–T as prespecified contrasts. The second planning protocol also specified 1-, 5-, and 10-step rollout decoding and goal-difficulty stratification by room crossing and distance.

Evaluation-device amendments were recorded before the respective cohort outcomes: canonical planning evaluation used an RTX 3060, uniformly within each fresh cohort, while training used an RTX 5090. One interrupted run was resumed from a full training checkpoint.

C.3 Seeding and pairing

The launch scripts pass the nominal seeds listed above, but the archived training logs also report a training-manager seed of `None` followed by a default seed of zero. These messages do not establish identical initial weights or data orders. The records do not verify distinct global random states across runs or matching initial states within J/B pairs. We therefore treat runs as unpaired; the reported Welch intervals use an independent-run approximation, with this unresolved seeding issue as an additional limitation.

At fixed inputs, the recorded J/B implementation check reports loss agreement within 10^{-10} and different encoder gradients. This supports the intended local computational intervention; it does not establish matching initial states or explain the full training outcomes.

C.4 Uncertainty in differences between configurations

To expose the uncertainty hidden by point-estimate rankings, we compute Welch intervals from the recorded model-level scores. These are post-hoc summaries, not a new prespecified testing procedure. Let $\bar{x}_a$, s_a^2, and n_a be the sample mean, sample variance, and number of models in arm a, and define the analogous quantities for arm b. Writing $q_a = s_a^2/n_a$ and $q_b = s_b^2/n_b$, the estimated difference, standard error, approximate degrees of freedom, and interval are

$$\widehat{\Delta} = \bar{x}_a - \bar{x}_b, \qquad \mathrm{SE} = \sqrt{q_a + q_b}, \qquad \nu = \frac{(q_a + q_b)^2}{q_a^2/(n_a - 1) + q_b^2/(n_b - 1)}, \qquad \widehat{\Delta} \pm t_{\nu,0.975}\,\mathrm{SE}. \tag{38}$$

$t_{\nu,0.975}$ is the 97.5th percentile of Student's t distribution with ν degrees of freedom. Planning differences are in percentage points; Reacher differences are in R^2 units. We use the unrounded records and do not pair models by nominal seed label.

The calculation assumes independent runs and a sufficiently regular sampling distribution of the mean. Neither assumption is established by three to five planning runs, particularly when outcomes split into high- and low-performing groups. The unverified Two-Room seeding (Appendix C.3) is an additional limitation. These intervals are conditional on the fixed datasets and evaluation manifests and do not quantify dataset or task variation. They are unadjusted for the number of contrasts and support neither familywise significance claims nor equivalence claims.

Table 5 includes every protocol-specified contrast and the two exploratory second-cohort contrasts discussed in the main text. Intervals are not selectively omitted when poor outcomes occur; instead, all individual scores remain visible and the approximation's limitations apply to the entire table. For example, the B–J interval is wide and includes zero despite a 21.5-point observed difference. This is compatible with a promising configuration-level result, but does not resolve its reliability or identify a general collapse mechanism.

D Complete Reacher results and probe interpretation

D.1 Fresh runs and statistics-mode ablation

Table 6 reports all 50 fresh Reacher runs. Both threshold indicators are retained: $R^2_{\min} > 0.3$ requires both angles to be decoded, whereas $R^2 > 0.6$ can be satisfied by an uneven allocation of performance.

Table 5: **Post-hoc Welch 95% intervals under an independent-run approximation.** Status refers to the contrast's place in the frozen protocol, not to prespecification of this interval procedure. The n_a/n_b column gives model counts. Read these intervals with the sampling, seeding, and multiplicity qualifications above.

Cohort	Outcome	Contrast	Status	n_a/n_b	Difference	Interval
Reacher	R^2	I–M	Primary	10/10	0.251	$[0.147, 0.355]$
Reacher	R^2	I–N	Primary	10/10	0.285	$[0.178, 0.393]$
Reacher	R^2	I–T	Primary	10/10	0.321	$[0.231, 0.410]$
Reacher	$R^2_{\min}$	I–M	Primary	10/10	0.451	$[0.242, 0.661]$
Reacher	$R^2_{\min}$	I–N	Primary	10/10	0.490	$[0.279, 0.702]$
Reacher	$R^2_{\min}$	I–T	Primary	10/10	0.646	$[0.466, 0.826]$
First	SR	G–M	Primary	5/5	4.00	$[2.56, 5.44]$
First	SR	G–U	Primary	5/5	1.30	$[-0.21, 2.81]$
First	SR	G–S	Primary	5/5	1.30	$[-1.55, 4.15]$
First	SR	I–Z	Secondary	5/5	29.20	$[-0.95, 59.35]$
First	SR	G–H	Secondary	5/5	0.80	$[-1.65, 3.25]$
First	SR	I–M	Secondary	5/5	-8.00	$[-38.16, 22.16]$
Second	SR	T–G	Primary	3/3	-7.67	$[-64.32, 48.99]$
Second	SR	G–C	Primary	3/3	40.83	$[-8.54, 90.20]$
Second	SR	T–C	Primary	3/3	33.17	$[-16.05, 82.39]$
Second	SR	B–J	Primary	3/3	21.50	$[-21.49, 64.49]$
Second	SR	G–M	Exploratory	3/3	5.67	$[1.78, 9.56]$
Second	SR	B–M	Exploratory	3/3	10.50	$[6.80, 14.20]$

The continuous values remain the primary evidence. The statistics-mode ablation in Table 7 shows the benchmark-specific reversal without mixing these nine runs with the main cohort.

Table 6: **All fresh Reacher runs.** Coordinate 1 is the shoulder and coordinate 2 is the wrist. Indicator columns use strict inequalities.

Seed	R^2	R^2_1	R^2_2	$R^2_{\min}$	Both > 0.3	Mean > 0.6
$\delta = 16$, *Innovation SIGReg*						
2001	0.693	0.601	0.784	0.601	1	1
2002	0.519	0.276	0.762	0.276	0	0
2003	0.438	0.110	0.765	0.110	0	0
2004	0.426	0.105	0.748	0.105	0	0
2005	0.474	0.203	0.746	0.203	0	0
$\delta = 16$, *Marginal SIGReg*						
2001	0.600	0.498	0.701	0.498	1	0
2002	0.401	0.638	0.164	0.164	0	0
2003	0.318	0.637	-0.002	-0.002	0	0
2004	0.179	0.698	-0.340	-0.340	0	0
2005	0.356	0.240	0.473	0.240	0	0
$\delta = 64$, *Innovation SIGReg*						
2001	0.842	0.783	0.901	0.783	1	1
2002	0.841	0.779	0.902	0.779	1	1
2003	0.827	0.739	0.914	0.739	1	1
2004	0.817	0.728	0.906	0.728	1	1
2005	0.840	0.779	0.900	0.779	1	1
2006	0.852	0.801	0.903	0.801	1	1
2007	0.460	0.012	0.908	0.012	0	0
2008	0.850	0.800	0.900	0.800	1	1
2009	0.855	0.799	0.911	0.799	1	1
2010	0.691	0.465	0.917	0.465	1	1
$\delta = 64$, *Marginal SIGReg*						
2001	0.750	0.639	0.860	0.639	1	1
2002	0.479	0.079	0.879	0.079	0	0
2003	0.525	0.177	0.873	0.177	0	0
2004	0.497	0.122	0.872	0.122	0	0

Table 6 continued

Seed	R^2	R_1^2	R_2^2	$R_{\min}^2$	Both > 0.3	Mean > 0.6
2005	0.603	0.365	0.840	0.365	1	1
2006	0.588	0.316	0.859	0.316	1	0
2007	0.444	0.058	0.830	0.058	0	0
2008	0.454	0.036	0.872	0.036	0	0
2009	0.551	0.258	0.843	0.258	0	0
2010	0.476	0.120	0.831	0.120	0	0
$\delta = 64$, *Normalization only*						
2001	0.552	0.286	0.818	0.286	0	0
2002	0.480	0.091	0.869	0.091	0	0
2003	0.417	0.042	0.793	0.042	0	0
2004	0.770	0.673	0.867	0.673	1	1
2005	0.447	0.091	0.803	0.091	0	0
2006	0.498	0.171	0.824	0.171	0	0
2007	0.515	0.209	0.821	0.209	0	0
2008	0.454	0.141	0.767	0.141	0	0
2009	0.435	0.010	0.859	0.010	0	0
2010	0.452	0.068	0.837	0.068	0	0
$\delta = 64$, *Temporal difference*						
2001	0.464	0.020	0.907	0.020	0	0
2002	0.462	0.019	0.905	0.019	0	0
2003	0.460	0.018	0.901	0.018	0	0
2004	0.468	0.023	0.912	0.023	0	0
2005	0.470	0.026	0.913	0.026	0	0
2006	0.469	0.032	0.905	0.032	0	0
2007	0.468	0.021	0.915	0.021	0	0
2008	0.474	0.024	0.923	0.024	0	0
2009	0.465	0.022	0.908	0.022	0	0
2010	0.468	0.023	0.913	0.023	0	0

Table 7: **Reacher statistics-mode comparison, every run.** All configurations share stride 64, $\lambda = 0.005$, and $w_{\mathrm{m}} = 0.1$.

Statistics mode	Seed	R^2	$R_{\min}^2$
EMA-detached	5201	0.842	0.775
EMA-detached	5202	0.811	0.721
EMA-detached	5203	0.846	0.785
Batch-detached	5201	0.009	0.002
Batch-detached	5202	0.018	0.000
Batch-detached	5203	0.004	-0.000
Batch-differentiable	5201	0.427	0.002
Batch-differentiable	5202	0.424	0.005
Batch-differentiable	5203	0.411	0.003

D.2 What the joint-angle probe establishes

The fitting and scoring distributions of this probe differ. At stride 64, the recorded shoulder-angle span of the fitting set is approximately 14.1 radians, with 10.9% of observations more than π from its median. The Gaussian-angle scoring set has far less mass in this range (approximately 0.155%). Consequently, a good test score establishes a useful linear readout under this particular distribution shift. It does not resolve the periodic ambiguity of raw angles across arbitrary trajectories or verify the full state, which would include variables beyond the two probed angles.

Section 4.2 reports a shuffled-label control. A linear map from true angles to sine/cosine targets scores -0.4289 under the same fit/test distributions. Since the targets are nonlinear functions of the angles, this failure is compatible with a perfect angle representation. The learned representations' sine/cosine scores are therefore retained as measurements (Table 8), but are not used to infer loss of all angular information.

A separate probe from wrapped angles to raw angles scores 0.8149. This is a reference for that feature parameterization and linear fitting procedure. It is not an information-theoretic upper bound: a different representation or decoder can have different approximation error. No percentage-of-ceiling claim follows from comparing Innovation SIGReg's score with this reference. Because the shoulder joint is unbounded, configurations differing by 2π render identically; since the fitting set spans about 14.1 radians, the raw-angle target is not a function of the image for part of the fitting data. This ambiguity is inherited from the benchmark and limits any linear probe fitted on these data. The wrapped-angle reference is likewise not an upper bound: the EMA-detached models in Table 7 exceed it on average. The wrist is limited to ± 2.79 rad and is unaffected.

Table 8: **Recorded sine/cosine-target linear probes.** Mean $\pm$ sample SD across models. Negative values are retained. These scores describe the particular nonlinear-target/linear-decoder combination and are not used as a general identification criterion.

Stride	Arm	n	Sine/cosine R^2
16	Innovation SIGReg	5	0.083 ± 0.275
16	Marginal SIGReg	5	0.083 ± 0.271
64	Innovation SIGReg	10	-0.250 ± 0.154
64	Marginal SIGReg	10	-0.009 ± 0.230
64	Normalization only	10	-0.051 ± 0.116
64	Temporal difference	10	0.131 ± 0.005

D.3 Development sweeps and the role of the anchor

Table 9 summarizes all non-smoke cells in the 335-record development registry; the two smoke records remain in the accompanying CSV but are excluded from the table. Each row is a single configuration, and its SD is computed over all recorded runs in that cell. The unequal cell sizes reflect development-stage expansion of selected settings. They do not constitute a balanced hyperparameter benchmark and are not pooled with fresh confirmation results.

The development comparison at stride 64 gives 0.755 ± 0.154 for the selected innovation cell and 0.545 ± 0.071 for the selected marginal cell, with 8/10 and 5/10 two-coordinate successes. The corresponding stride-16 values are 0.558 ± 0.106 and 0.405 ± 0.193, with 6/10 and 5/10 successes. The stronger anchor used at stride 16 therefore belongs to the selected configuration; it cannot be omitted when interpreting the stride comparison. The three-run $\lambda = 0.005$, $w_{\mathrm{m}} = 1$ cell reaches a higher mean, 0.618 ± 0.157, than the selected ten-run setting. Full-matrix, clamp, anchor, differentiable-statistics, and extended-duration variants are shown separately below. Their results do not establish that any one setting is universally optimal. In particular, the 200-epoch experiment also changes the optimization schedule, so it does not isolate additional training time at a fixed schedule.

The fixed-λ examples promoted to Section 4.2 use $\lambda = 0.005$. At stride 16, the light-anchor, strong-anchor, and full-matrix cells each contain three models. The full-matrix launch record specifies $w_{\mathrm{m}} = 1$: its appropriate transition-model comparison is therefore the strong-anchor diagonal cell, not the light-anchor cell. It also changes residual whitening from coordinatewise scaling to a covariance inverse square root, as described in the method. Its mean 0.565 versus 0.618 does not support an improvement in this cell, but three models do not establish equivalence or global superiority of the diagonal estimator. The stride-64 anchor comparison has unequal cell sizes, ten versus three, because the light-anchor cell was expanded during development. These comparisons illustrate configuration sensitivity and remain separate from the fresh cohorts.

Table 9: **Complete development-cell summaries.** M: marginal; I: innovation. "Traj." uses stride δ; OU uses transition coefficient ρ. Standard innovation cells have the light anchor; the listed variants modify that setting. "Both" counts $R^2_{\min} > 0.3$. These are historical, selected and expanded cells.

Configuration	Data	δ/ρ	Arm	λ	n	R^2	Both
Clamp 0.9	Traj.	16	I	0.001	3	0.405 ± 0.001	0/3
Clamp 0.9	Traj.	16	I	0.005	3	0.382 ± 0.022	0/3

Table 9 continued

Configuration	Data	δ/ρ	Arm	λ	n	R^2	Both
Anchor 1 (diagnostic)	Traj.	16	I	0.005	3	0.619 ± 0.157	2/3
Anchor 1 (diagnostic)	Traj.	16	I	0.05	3	0.244 ± 0.049	0/3
Standard	OU	0.95	M	0.001	3	0.925 ± 0.006	3/3
Standard	OU	0.95	M	0.005	3	0.934 ± 0.002	3/3
Standard	OU	0.95	M	0.01	3	0.942 ± 0.003	3/3
Standard	OU	0.95	M	0.05	3	0.950 ± 0.000	3/3
Standard	OU	0.99	M	0.001	3	0.946 ± 0.000	3/3
Standard	OU	0.99	M	0.005	3	0.949 ± 0.000	3/3
Standard	OU	0.99	M	0.01	3	0.949 ± 0.001	3/3
Standard	OU	0.99	M	0.05	3	0.437 ± 0.106	1/3
Standard	Traj.	1	I	0.001	3	0.251 ± 0.122	0/3
Standard	Traj.	1	I	0.005	3	-0.066 ± 0.018	0/3
Standard	Traj.	1	I	0.01	3	0.205 ± 0.038	0/3
Standard	Traj.	1	I	0.05	3	0.143 ± 0.052	0/3
Standard	Traj.	1	M	0.001	3	0.163 ± 0.063	0/3
Standard	Traj.	1	M	0.005	3	0.289 ± 0.008	0/3
Standard	Traj.	1	M	0.01	3	0.278 ± 0.111	0/3
Standard	Traj.	1	M	0.05	3	0.153 ± 0.108	0/3
Standard	Traj.	2	I	0.001	3	0.392 ± 0.002	0/3
Standard	Traj.	2	I	0.005	3	0.383 ± 0.021	0/3
Standard	Traj.	2	I	0.01	3	0.265 ± 0.049	0/3
Standard	Traj.	2	I	0.05	3	0.233 ± 0.102	0/3
Standard	Traj.	2	M	0.001	3	0.394 ± 0.026	0/3
Standard	Traj.	2	M	0.005	3	0.325 ± 0.105	0/3
Standard	Traj.	2	M	0.01	3	0.334 ± 0.060	0/3
Standard	Traj.	2	M	0.05	3	0.171 ± 0.166	0/3
Standard	Traj.	8	I	0.001	3	0.409 ± 0.001	0/3
Standard	Traj.	8	I	0.005	3	0.346 ± 0.061	0/3
Standard	Traj.	8	I	0.01	3	0.375 ± 0.034	0/3
Standard	Traj.	8	I	0.05	3	0.273 ± 0.042	0/3
Standard	Traj.	8	M	0.001	3	0.470 ± 0.004	0/3
Standard	Traj.	8	M	0.005	3	0.539 ± 0.078	2/3
Standard	Traj.	8	M	0.01	3	0.522 ± 0.103	1/3
Standard	Traj.	8	M	0.05	3	0.321 ± 0.031	0/3
Standard	Traj.	16	I	0.001	3	0.404 ± 0.002	0/3
Standard	Traj.	16	I	0.005	3	0.392 ± 0.015	0/3
Standard	Traj.	16	I	0.01	3	0.377 ± 0.028	0/3
Standard	Traj.	16	I	0.05	3	0.258 ± 0.084	0/3
Standard	Traj.	16	M	0.001	3	0.465 ± 0.009	0/3
Standard	Traj.	16	M	0.005	3	0.444 ± 0.007	0/3
Standard	Traj.	16	M	0.01	3	0.537 ± 0.156	1/3
Standard	Traj.	16	M	0.05	10	0.405 ± 0.193	5/10
Standard	Traj.	64	I	0.001	3	0.322 ± 0.065	0/3
Standard	Traj.	64	I	0.005	10	0.755 ± 0.154	8/10
Standard	Traj.	64	I	0.01	3	0.468 ± 0.001	0/3
Standard	Traj.	64	I	0.05	3	0.488 ± 0.001	0/3
Standard	Traj.	64	M	0.001	3	0.291 ± 0.002	0/3
Standard	Traj.	64	M	0.005	3	0.368 ± 0.012	0/3
Standard	Traj.	64	M	0.01	3	0.421 ± 0.009	0/3
Standard	Traj.	64	M	0.05	10	0.545 ± 0.071	5/10
200 epochs	Traj.	64	I	0.005	10	0.486 ± 0.160	3/10
Full matrix	Traj.	8	I	0.001	3	0.578 ± 0.209	1/3
Full matrix	Traj.	8	I	0.005	3	0.453 ± 0.167	1/3
Full matrix	Traj.	8	I	0.01	3	0.276 ± 0.034	0/3
Full matrix	Traj.	8	I	0.05	3	0.286 ± 0.083	0/3
Full matrix	Traj.	16	I	0.001	3	0.473 ± 0.009	0/3
Full matrix	Traj.	16	I	0.005	3	0.565 ± 0.104	2/3
Full matrix	Traj.	16	I	0.01	3	0.500 ± 0.054	1/3
Full matrix	Traj.	16	I	0.05	3	0.252 ± 0.029	0/3
Marginal, extra weights	Traj.	16	M	0.02	3	0.457 ± 0.027	0/3
Marginal, extra weights	Traj.	16	M	0.1	3	0.331 ± 0.072	0/3
Marginal, extra weights	Traj.	64	M	0.02	3	0.473 ± 0.017	0/3

Table 9 continued

Configuration	Data	δ/ρ	Arm	λ	n	R^2	Both
Marginal, extra weights	Traj.	64	M	0.1	10	0.547 ± 0.101	4/10
Anchor 0.1, instrumented	Traj.	16	I	0.01	3	0.379 ± 0.017	0/3
Anchor 0.1, instrumented	Traj.	64	I	0.01	3	0.471 ± 0.002	0/3
No anchor, instrumented	Traj.	16	I	0.01	3	0.339 ± 0.032	0/3
No anchor, instrumented	Traj.	64	I	0.01	3	0.311 ± 0.041	0/3
Anchor 1	Traj.	1	I	0.001	3	0.154 ± 0.110	0/3
Anchor 1	Traj.	1	I	0.005	3	0.126 ± 0.098	0/3
Anchor 1	Traj.	1	I	0.01	3	0.065 ± 0.125	0/3
Anchor 1	Traj.	1	I	0.05	3	0.066 ± 0.107	0/3
Anchor 1	Traj.	2	I	0.001	3	0.359 ± 0.055	0/3
Anchor 1	Traj.	2	I	0.005	3	0.242 ± 0.146	0/3
Anchor 1	Traj.	2	I	0.01	3	0.157 ± 0.152	0/3
Anchor 1	Traj.	2	I	0.05	3	0.153 ± 0.059	0/3
Anchor 1	Traj.	8	I	0.001	3	0.584 ± 0.211	1/3
Anchor 1	Traj.	8	I	0.005	3	0.520 ± 0.105	2/3
Anchor 1	Traj.	8	I	0.01	3	0.296 ± 0.030	0/3
Anchor 1	Traj.	8	I	0.05	3	0.289 ± 0.010	0/3
Anchor 1	Traj.	16	I	0.001	3	0.470 ± 0.008	0/3
Anchor 1	Traj.	16	I	0.005	3	0.618 ± 0.157	2/3
Anchor 1	Traj.	16	I	0.01	10	0.558 ± 0.106	6/10
Anchor 1	Traj.	16	I	0.05	3	0.239 ± 0.061	0/3
Anchor 1	Traj.	64	I	0.001	3	0.314 ± 0.014	0/3
Anchor 1	Traj.	64	I	0.005	3	0.444 ± 0.011	0/3
Anchor 1	Traj.	64	I	0.01	3	0.474 ± 0.007	0/3
Anchor 1	Traj.	64	I	0.05	3	0.440 ± 0.013	0/3
Batch-differentiable	Traj.	16	I	0.001	3	0.402 ± 0.006	0/3
Batch-differentiable	Traj.	16	I	0.005	3	0.408 ± 0.001	0/3
Batch-differentiable	Traj.	16	I	0.01	3	0.379 ± 0.009	0/3
Batch-differentiable	Traj.	16	I	0.05	3	0.271 ± 0.088	0/3
Batch-differentiable	Traj.	64	I	0.001	3	0.094 ± 0.069	0/3
Batch-differentiable	Traj.	64	I	0.005	3	0.428 ± 0.008	0/3
Batch-differentiable	Traj.	64	I	0.01	3	0.458 ± 0.001	0/3
Batch-differentiable	Traj.	64	I	0.05	3	0.490 ± 0.010	0/3

D.4 True-state residual diagnostics

The state-level diagnostic fits transition estimators on one half of the episodes and evaluates them on the other. Stride-spaced pairs avoid overlapping multi-step transition intervals. Table 10 compares a pooled, shared-normalization diagonal estimate (P), coordinatewise least-squares transitions with intercepts (A), and a full linear transition with an intercept (F). A and F use residual scales estimated on the fitting episodes. These are diagnostics on true angles, not oracle innovations or a reproduction of every training normalizer.

At stride 1, fitted residual excess kurtosis is approximately -0.56; at strides 16 and 64 it is closer to zero. Nonetheless, the pooled estimate retains linear residual–state correlation, including 0.208 at stride 64, and the absolute-value correlations remain nonzero after least-squares fitting. Near-Gaussian one-dimensional shape therefore does not verify independent transition noise. In particular, residual histograms alone cannot establish the assumptions motivating the method.

E Complete planning results and additional environments

E.1 The two main Two-Room cohorts

Tables 11 and 12 give every evaluation of every model in the two main cohorts, including all poor outcomes. A low-SR model tends to perform poorly across evaluation seeds; its contribution is retained in the all-run mean. The separate tables preserve the distinction between the first design and its subsequently specified second cohort. No inferential claim treats the four evaluation rows of a model as independent training replications.

Table 10: **True-angle residual diagnostics on test episodes.** Variance and excess kurtosis are averaged over coordinates. c is the largest absolute same-coordinate residual–state correlation; c_{abs} replaces both variables by their absolute values. These summaries are not independence tests.

Stride	Estimator	Variance	Excess kurtosis	c	c_{abs}
1	P	0.180	-0.548	0.252	0.161
1	A	0.998	-0.559	0.001	0.164
1	F	0.998	-0.559	0.001	0.164
2	P	0.472	-0.347	0.158	0.091
2	A	0.996	-0.348	0.001	0.091
2	F	0.996	-0.348	0.001	0.090
8	P	1.001	-0.093	0.092	0.076
8	A	0.995	-0.091	0.003	0.077
8	F	0.995	-0.091	0.003	0.077
16	P	1.013	-0.044	0.117	0.092
16	A	1.001	-0.043	0.004	0.092
16	F	1.001	-0.043	0.004	0.092
64	P	1.055	-0.043	0.208	0.166
64	A	1.008	-0.053	0.011	0.166
64	F	1.008	-0.053	0.011	0.166

Table 11: **First Two-Room cohort, complete evaluation scores.** The second column contains nominal training-seed labels. The four evaluation columns are SR in percent; the final column is their mean.

Arm	Label	Eval 0	Eval 1	Eval 2	Eval 42	Model SR
M	1001	84.0	94.0	88.0	94.0	90.0
M	1002	82.0	92.0	84.0	94.0	88.0
M	1003	86.0	92.0	82.0	92.0	88.0
M	1004	80.0	94.0	88.0	94.0	89.0
M	1005	84.0	94.0	82.0	94.0	88.5
I	1001	98.0	98.0	98.0	98.0	98.0
I	1002	60.0	64.0	60.0	48.0	58.0
I	1003	96.0	98.0	98.0	98.0	97.5
I	1004	46.0	50.0	56.0	50.0	50.5
I	1005	100.0	100.0	100.0	98.0	99.5
H	1001	86.0	96.0	86.0	92.0	90.0
H	1002	86.0	96.0	88.0	92.0	90.5
H	1003	90.0	98.0	94.0	96.0	94.5
H	1004	88.0	98.0	92.0	96.0	93.5
H	1005	82.0	100.0	88.0	94.0	91.0
S	1001	88.0	96.0	96.0	96.0	94.0
S	1002	76.0	96.0	90.0	90.0	88.0
S	1003	86.0	96.0	88.0	96.0	91.5
S	1004	88.0	98.0	92.0	94.0	93.0
S	1005	86.0	92.0	92.0	92.0	90.5
G	1001	84.0	100.0	90.0	90.0	91.0
G	1002	86.0	98.0	96.0	96.0	94.0
G	1003	86.0	100.0	92.0	92.0	92.5
G	1004	88.0	96.0	92.0	96.0	93.0
G	1005	86.0	98.0	92.0	96.0	93.0
Z	1001	52.0	48.0	56.0	48.0	51.0
Z	1002	44.0	56.0	56.0	50.0	51.5
Z	1003	44.0	50.0	58.0	54.0	51.5
Z	1004	40.0	48.0	58.0	54.0	50.0
Z	1005	48.0	56.0	58.0	52.0	53.5
U	1001	92.0	94.0	90.0	94.0	92.5
U	1002	84.0	94.0	88.0	94.0	90.0
U	1003	84.0	98.0	90.0	92.0	91.0
U	1004	92.0	94.0	90.0	90.0	91.5

Table 11 continued

Arm	Label	Eval 0	Eval 1	Eval 2	Eval 42	Model SR
U	1005	92.0	96.0	90.0	90.0	92.0
T	1001	92.0	100.0	98.0	94.0	96.0
T	1002	98.0	100.0	100.0	96.0	98.5
T	1003	100.0	98.0	100.0	98.0	99.0

Table 12: **Second Two-Room cohort, complete evaluation scores.** Labels have the same nominal-seed interpretation as in the first cohort.

Arm	Label	Eval 0	Eval 1	Eval 2	Eval 42	Model SR
M	4101	84.0	94.0	84.0	94.0	89.0
M	4102	80.0	90.0	82.0	90.0	85.5
M	4103	82.0	90.0	82.0	92.0	86.5
G	4101	86.0	98.0	94.0	90.0	92.0
G	4102	88.0	100.0	94.0	96.0	94.5
G	4103	88.0	94.0	88.0	96.0	91.5
T	4101	60.0	62.0	48.0	64.0	58.5
T	4102	98.0	100.0	100.0	98.0	99.0
T	4103	96.0	100.0	100.0	94.0	97.5
C	4101	66.0	70.0	78.0	70.0	71.0
C	4102	28.0	24.0	42.0	30.0	31.0
C	4103	56.0	52.0	52.0	54.0	53.5
J	4101	56.0	56.0	58.0	60.0	57.5
J	4102	88.0	86.0	74.0	66.0	78.5
J	4103	92.0	86.0	96.0	94.0	92.0
B	4101	96.0	100.0	100.0	98.0	98.5
B	4102	96.0	100.0	98.0	96.0	97.5
B	4103	96.0	100.0	96.0	94.0	96.5

The second cohort's guard-minus-marginal difference is 5.67 points; its differencing-minus-guard difference is -7.67 points. The earlier guard advantage over fixed and scheduled anchors remains small relative to these between-cohort changes in differencing. Together the cohorts support the guard as a useful tested stabilization configuration, without establishing adaptive triggering as the essential ingredient. TC-LeWM's comparator changes centering, residual scaling, and the anchor together; its result is not a clean ablation of any one of those components. The executed four-frame window agrees with the reported window of TC-LeWM version 3 (Liu et al., 2026).

E.2 Official data, PushT, and preliminary Cube evaluation

Table 13: **Additional settings for the earlier unguarded configuration.** n counts trained models; SDs summarize variation across models. PushT's marginal row has only one model, so its evaluation-seed SD of 2.3 is not included as training variability. These development cohorts remain separate from the main comparisons.

Setting	Marginal SR (%)	n	Innovation SR (%)	n
Two-Room, official data	87.7 ± 1.6	3	83.8 ± 20.8	3
PushT, local cohort	86.0	1	79.0 ± 7.5	3

The official Two-Room experiment trains on the 10,000-episode dataset, separately from the regenerated-data cohorts. It evaluates the earlier batch-differentiable configuration. Its all-run mean remains below the marginal baseline, with scores 93.0, 60.0, and 98.5 across its three runs. Table 14 provides the underlying evaluations. This supports recurrence of poor outcomes in that configuration on a second dataset; it does not evaluate the later batch-detached variant.

Table 14: **Official Two-Room data. The `nodetach` record label denotes batch-differentiable Innovation SIGReg.** Final-column values are mean ± sample SD over the four evaluations of one model.

Configuration	Label	Eval 0	Eval 1	Eval 2	Eval 42	Eval mean ± SD
marginal	3072	86.0	92.0	84.0	86.0	87.00 ± 3.46
marginal	0	84.0	90.0	84.0	88.0	86.50 ± 3.00
marginal	1	84.0	90.0	94.0	90.0	89.50 ± 4.12
nodetach	3072	90.0	90.0	98.0	94.0	93.00 ± 3.83
nodetach	0	66.0	52.0	58.0	64.0	60.00 ± 6.32
nodetach	1	96.0	100.0	100.0	98.0	98.50 ± 1.91

PushT's local innovation cohort has model-level SR values 79.0, 71.5, and 86.5, with mean $79.0 \pm 7.5\%$ (Table 15). The marginal reference is one trained model with mean SR 86.0% and SD 2.3 across evaluations. Under the development protocol's descriptive criterion of more than five points below that reference, two innovation runs regress. This threshold is specific to this comparison and is not interchangeable with a universal "collapse" label.

Table 15: **PushT, local cohort. The final-column SD is across four evaluation seeds for one model, not across training runs.** Final-column values are mean ± sample SD over the four evaluations of one model.

Configuration	Label	Eval 0	Eval 1	Eval 2	Eval 42	Eval mean ± SD
innovation	3072	76.0	82.0	80.0	78.0	79.00 ± 2.58
innovation	0	74.0	72.0	68.0	72.0	71.50 ± 2.52
innovation	1	90.0	92.0	80.0	84.0	86.50 ± 5.51
marginal	3072	88.0	84.0	84.0	88.0	86.00 ± 2.31

Reruns of the same nominal innovation settings on H100 GPUs are reported separately in Table 16. Their SR values differ substantially from the local results, including $79.0 \rightarrow 8.0$ for label 3072 and $71.5 \rightarrow 82.0$ for label 0. These observations establish differences across recorded hardware/software executions; they do not isolate a hardware effect or establish that initialization has no influence. The cohorts are not pooled. The corresponding raw participation ratios are 5.84, 5.21, and 4.77, compared with local values 4.92, 3.76, and 8.93. Even their ordering does not provide a transferable prediction of SR.

Table 16: **PushT, historical H100 reruns. Evaluation-seed SDs are shown separately for each model.** Final-column values are mean ± sample SD over the four evaluations of one model.

Configuration	Label	Eval 0	Eval 1	Eval 2	Eval 42	Eval mean ± SD
innovation	3072	6.0	6.0	10.0	10.0	8.00 ± 2.31
innovation	0	80.0	90.0	76.0	82.0	82.00 ± 5.89
innovation	1	74.0	76.0	70.0	82.0	75.50 ± 5.00

The Cube experiment contains only one trained model per arm (Table 17). Innovation achieves 68.0% and marginal SIGReg 59.0%, a nine-point difference. The respective SDs of 5.9 and 3.8 summarize evaluation-seed variation. The record does not estimate training variability and cannot establish robustness or broad transfer of the method. Raw participation ratios of approximately 11.3 and 42.7 further illustrate why a lower variance-spread measure need not imply worse behavior in every setting.

Table 17: **Cube, one training run per arm. The `nodetach` label denotes the earlier batch-differentiable innovation configuration.** Final-column values are mean ± sample SD over the four evaluations of one model.

Configuration	Label	Eval 0	Eval 1	Eval 2	Eval 42	Eval mean ± SD
marginal	3072	62.0	54.0	58.0	62.0	59.00 ± 3.83
nodetach	3072	76.0	62.0	66.0	68.0	68.00 ± 5.89

E.3 Development cohorts, selected rescues, and sensitivity

The development experiments explain the selection of later controls, but do not augment their confirmation sample sizes. Table 18 retains the original marginal, EMA-detached innovation, and batch-differentiable innovation cohorts. Their all-run means are $89.0 \pm 2.2\%$, $84.7 \pm 9.8\%$, and $86.1 \pm 17.4\%$, respectively. The different run counts and adaptive development history preclude treating this table as a balanced, prospectively specified statistics-mode experiment. The label-3072 differentiable model is reused in the configuration probe below.

Table 18: **Historical regenerated-data Two-Room cohorts. `innovation` uses EMA-detached statistics; `batch_differentiable` uses current-batch differentiable statistics.** Final-column values are mean ± sample SD over the four evaluations of one model.

Configuration	Label	Eval 0	Eval 1	Eval 2	Eval 42	Eval mean ± SD
marginal	3072	84.0	92.0	82.0	94.0	88.00 ± 5.89
marginal	0	86.0	98.0	86.0	96.0	91.50 ± 6.40
marginal	1	84.0	88.0	86.0	92.0	87.50 ± 3.42
innovation	3072	74.0	78.0	72.0	92.0	79.00 ± 9.02
innovation	0	94.0	98.0	96.0	96.0	96.00 ± 1.63
innovation	1	70.0	86.0	74.0	86.0	79.00 ± 8.25
batch_differentiable	3072	94.0	98.0	98.0	94.0	96.00 ± 2.31
batch_differentiable	0	98.0	100.0	100.0	96.0	98.50 ± 1.91
batch_differentiable	1	94.0	88.0	96.0	92.0	92.50 ± 3.42
batch_differentiable	2	54.0	46.0	60.0	60.0	55.00 ± 6.63
batch_differentiable	3	92.0	98.0	100.0	94.0	96.00 ± 3.65
batch_differentiable	4	98.0	100.0	100.0	98.0	99.00 ± 1.15
batch_differentiable	5	92.0	88.0	86.0	92.0	89.50 ± 3.00
batch_differentiable	6	64.0	62.0	58.0	64.0	62.00 ± 2.83

An initial probe of the poor EMA-detached run with nominal label 3072 scored 54.0% without the anchor, 45.5% at outer weight 0.03, and 83.5% with a fixed anchor of 0.5. Switching to batch-differentiable statistics scored 96.0%. These single-run probes motivated later experiments; they neither prove that an anchor excludes collapse nor separate freshness from gradient flow. The latter distinction is addressed by the three-mode and J/B comparisons in the main text.

Table 19: **Single-run historical configuration probes on nominal label 3072. `mw0`/`mw05`: anchor weights 0/0.5; `lam03`: outer weight 0.03; `nodetach`: batch-differentiable statistics.** Final-column values are mean ± sample SD over the four evaluations of one model.

Configuration	Label	Eval 0	Eval 1	Eval 2	Eval 42	Eval mean ± SD
mw0	3072	56.0	46.0	64.0	50.0	54.00 ± 7.83
mw05	3072	78.0	88.0	78.0	90.0	83.50 ± 6.40
lam03	3072	44.0	46.0	48.0	44.0	45.50 ± 1.91
nodetach	3072	94.0	98.0	98.0	94.0	96.00 ± 2.31

The six-run historical guard cohort was partly selected after outcomes were known: it includes two observed poor runs and four successful controls. Its unguarded mean is $84.4 \pm 20.2\%$, compared with

$92.4 \pm 1.7\%$ after guarded retraining (Table 20). The two selected poor outcomes change from 55.0 to 94.5 and from 62.0 to 90.5. The four successful controls lose five points on average. Selection and regression to the mean can contribute to the apparent rescue; these runs are not combined with the two later cohorts.

Table 20: **Selected historical guard cohort. Every model uses the online threshold 12 and anchor boost to 0.5.** Final-column values are mean ± sample SD over the four evaluations of one model.

Configuration	Label	Eval 0	Eval 1	Eval 2	Eval 42	Eval mean ± SD
guard	0	88.0	98.0	86.0	92.0	91.00 ± 5.29
guard	2	88.0	100.0	94.0	96.0	94.50 ± 5.00
guard	6	80.0	98.0	90.0	94.0	90.50 ± 7.72
guard	3072	82.0	98.0	92.0	94.0	91.50 ± 6.81
guard	3	86.0	100.0	92.0	94.0	93.00 ± 5.77
guard	4	88.0	98.0	94.0	96.0	94.00 ± 4.32

On the same selected Two-Room label, thresholds 6, 12, and 20 give 93.5, 94.5, and 95.0% SR. The endpoint thresholds are shown in Table 21; threshold 12 appears above. This is limited sensitivity evidence on one selected label, not a general threshold-robustness result. The selected PushT guard run, with threshold 6, scores 88.5% instead of the earlier 71.5% (Table 22). It remains a single selected retraining result and does not establish cross-environment reliability.

Table 21: **Historical threshold sensitivity on nominal Two-Room label 2, with a common anchor boost to 0.5.** Final-column values are mean ± sample SD over the four evaluations of one model.

Configuration	Label	Eval 0	Eval 1	Eval 2	Eval 42	Eval mean ± SD
floor6	2	88.0	98.0	92.0	96.0	93.50 ± 4.43
floor20	2	88.0	100.0	96.0	96.0	95.00 ± 5.03

Table 22: **Selected PushT guard retraining, threshold 6.** Final-column values are mean ± sample SD over the four evaluations of one model.

Configuration	Label	Eval 0	Eval 1	Eval 2	Eval 42	Eval mean ± SD
guard	0	88.0	88.0	86.0	92.0	88.50 ± 2.52

E.4 When the stronger anchor begins

The fixed, scheduled, and guarded arms share batch-differentiable innovation statistics but introduce the stronger raw-marginal anchor at different times. H uses weight 0.5 throughout. S switches from 0.1 to 0.5 at update 4,960, fixed before the cohort as 8% of 62,000 planned updates. G switches once its online smoothed PR falls below 12 and then retains weight 0.5. The control isolates adaptive timing from the simpler explanation that a stronger anchor at an early training stage is sufficient for the observed improvement.

The six earlier trigger updates were 5,270, 5,703, 4,713, 4,183, 5,139, and 5,066, spanning 4,183–5,703 with mean 5,012.3. These historical observations informed the scheduled intervention; its update was not fitted to the fresh cohort's outcomes. Table 23 gives the actual guarded-cohort events parsed from the training logs. All eight models triggered. The first-cohort range is 3,978–5,469 and the second-cohort range is 4,442–4,936. Overlap with the scheduled update helps interpret the similar observed outcomes; it does not make adaptive and fixed-time interventions identical or establish their equivalence.

Table 23: **Recorded guard trigger events.** Update and online PR are read from each model's single trigger message. The threshold is 12 and the anchor increases from 0.1 to 0.5 in every row. Final SR is averaged over the model's four evaluations. These online PR values are distinct from the common-manifest offline measurements.

Cohort	Nominal label	Trigger update	Online PR	Final SR (%)
First	1001	4817	11.71	91.0
First	1002	5469	11.63	94.0
First	1003	4820	11.86	92.5
First	1004	3978	11.93	93.0
First	1005	4707	11.88	93.0
Second	4101	4499	11.99	92.0
Second	4102	4936	11.94	94.5
Second	4103	4442	11.76	91.5

F Representation geometry, state probes, and alarms

F.1 A common raw-embedding measurement

The offline geometry analysis uses a fixed manifest of 64 episodes with eight initial frames from each episode, selected with random seed 123 from the regenerated Two-Room dataset. All models are placed in evaluation mode, and their projected encoder outputs are collected on the same frames. Manifest episodes are drawn from the training dataset and may have been used to train each encoder. We therefore call this a common analysis manifest, rather than claiming it is an encoder-training-disjoint test set. There are 512 frame embeddings, with temporal dependence within episodes.

Let C be the sample covariance of the centered raw embeddings and $\ell_1, \ldots, \ell_d$ its nonnegative eigenvalues. For nonzero covariance, the participation ratio is

$$\mathrm{PR}(C) = \frac{(\sum_{j=1}^{d} \ell_j)^2}{\sum_{j=1}^{d} \ell_j^2}. \tag{39}$$

$d = 192$ for these models. PR equals one when a single covariance direction carries all variance, and increases as variance is spread more evenly. A value close to one does not prove that every smaller direction contains no information. The offline calculation uses raw embeddings, without coordinate standardization.

The guard's online monitor estimates raw covariance on training batches in training mode and smooths PR over updates with weights 0.95 and 0.05. It therefore differs from the offline measurement in input stream, network mode, and smoothing. Its threshold of 12 cannot be compared numerically with the offline alarm threshold of 2. Standardizing each embedding coordinate before covariance calculation produces another measure again; the recorded standardized values are not mixed into Figure 3.

The first cohort has raw PR ranges 28.860–30.468 for M, 13.839–15.999 for G, and 2.627–4.140 for T; Z has PR 1.000 in all five models. The three I models with SR above 97% have PR 1.965–3.369, while the two low-SR I models have PR 1.000–1.046. PR near one describes variance concentration; it does not by itself establish loss of all state information or a failed planner.

F.2 Position accessibility in both planning cohorts

The position probes partition the 64 manifest episodes using a fixed permutation with seed 777: 40 fit episodes, 12 validation episodes, and 12 test episodes. Within each partition the eight frames per episode are pooled. Thus the test comprises 96 correlated frame observations from 12 episodes, not 96 independent episodes. Feature means and standard deviations are fitted on the probe-training partition only. A constant feature is appended, and ridge coefficients in $\{10^{-4}, 10^{-3}, 10^{-2}, 10^{-1}, 1, 10\}$ are selected by validation mean squared error. The implemented ridge penalty includes the appended constant. No planning outcomes select the probe's regularization value.

We report R_x^2 and R_y^2 using Eq. equation 36, with agent position replacing angle, their unweighted mean $R_{\text{pos}}^2 = (R_x^2 + R_y^2)/2$, and root mean squared error (RMSE) over both coordinates in the dataset's recorded position units. The test episodes are disjoint from those used to fit and tune the probe, but are not established to be disjoint from encoder training. These are exploratory accessibility measurements rather than a broad out-of-distribution generalization test.

The 66-checkpoint probe registry covers all 38 first-cohort and 18 second-cohort models. Figure 3a,b uses those 56 models, marking cohorts separately; the remaining ten historical checkpoints are not pooled into the figure. Tables 24 and 25 preserve every plotted model and both coordinate scores.

Table 24: **All first-cohort geometry and position-probe records.** Definitions and manifest match the second-cohort table. SR is in percent; RMSE is in dataset position units.

Arm	Label	SR	PR	R_x^2	R_y^2	R_{pos}^2	RMSE
G	1001	91.0	15.510	0.784	0.478	0.631	28.03
G	1002	94.0	15.613	0.872	0.587	0.729	23.70
G	1003	92.5	13.839	0.723	0.738	0.730	25.20
G	1004	93.0	15.999	0.916	0.580	0.748	22.46
G	1005	93.0	15.107	0.894	0.662	0.778	21.49
H	1001	90.0	18.811	0.173	0.728	0.450	37.79
H	1002	90.5	18.894	0.845	0.532	0.688	25.50
H	1003	94.5	18.375	0.904	0.643	0.774	21.52
H	1004	93.5	17.820	0.536	0.555	0.546	32.69
H	1005	91.0	17.303	0.925	0.534	0.729	23.12
I	1001	98.0	2.666	0.934	0.985	0.959	10.41
I	1002	58.0	1.000	0.524	0.963	0.743	26.70
I	1003	97.5	3.369	0.867	0.730	0.798	20.93
I	1004	50.5	1.046	-0.474	0.773	0.150	48.05
I	1005	99.5	1.965	0.931	0.952	0.942	11.94
M	1001	90.0	28.860	0.704	0.393	0.549	31.26
M	1002	88.0	30.108	0.938	0.664	0.801	19.90
M	1003	88.0	29.197	0.967	0.627	0.797	19.72
M	1004	89.0	29.313	0.979	0.487	0.733	22.36
M	1005	88.5	30.468	0.961	0.299	0.630	26.40
S	1001	94.0	13.806	0.832	0.565	0.699	25.25
S	1002	88.0	12.386	0.966	0.417	0.692	24.13
S	1003	91.5	14.724	0.854	0.559	0.706	24.74
S	1004	93.0	14.254	0.877	0.587	0.732	23.54
S	1005	90.5	15.122	0.938	0.612	0.775	21.09
T	1001	96.0	4.140	0.991	0.975	0.983	6.02
T	1002	98.5	3.263	0.998	0.998	0.998	2.10
T	1003	99.0	2.627	0.999	0.996	0.998	2.20
U	1001	92.5	5.877	0.963	0.710	0.837	17.83
U	1002	90.0	5.581	0.972	0.972	0.972	8.10
U	1003	91.0	5.490	0.823	0.960	0.891	17.00
U	1004	91.5	6.551	0.898	0.769	0.834	18.88
U	1005	92.0	5.428	0.771	0.746	0.758	23.65
Z	1001	51.0	1.000	0.851	0.959	0.905	15.80
Z	1002	51.5	1.000	0.326	0.957	0.641	31.63
Z	1003	51.5	1.000	0.804	0.780	0.792	21.95
Z	1004	50.0	1.000	0.677	0.975	0.826	21.99
Z	1005	53.5	1.000	0.777	0.975	0.876	18.47

The J model with label 4101 combines PR approximately 1 with $R_{\text{pos}}^2 = 0.765$ and SR 57.5%. There is no mathematical contradiction: PR emphasizes variance concentration in raw coordinates, while the probe can rescale small directions. Moreover, recovering observed position does not establish accurate action-conditioned predictions or useful planning distances. The available records contain no completed rollout-decoding comparison that separates these possibilities, so we leave the remaining planning gap unexplained. TC-LeWM adaptation probes are negative in both coordinates for all three models, but this does not establish that all residual-family failures share that property.

Table 25: **All second-cohort geometry and position-probe records.** SR is the model-level planning percentage. PR is computed on raw embeddings; probe inputs are separately standardized from the probe-fit partition. RMSE is in dataset position units.

Arm	Label	SR	PR	R^2_x	R^2_y	R^2_{pos}	RMSE
B	4101	98.5	2.848	0.723	0.976	0.849	20.43
B	4102	97.5	2.940	0.867	0.992	0.929	14.04
B	4103	96.5	2.300	0.930	0.886	0.908	14.31
C	4101	71.0	15.040	-0.420	-0.536	-0.478	58.57
C	4102	31.0	2.259	-0.351	-0.672	-0.512	58.80
C	4103	53.5	9.003	-1.099	-2.231	-1.665	77.14
G	4101	92.0	14.515	0.821	0.721	0.771	22.60
G	4102	94.5	14.972	0.767	0.959	0.863	19.23
G	4103	91.5	15.259	0.894	0.807	0.850	18.11
J	4101	57.5	1.000	0.593	0.937	0.765	25.25
J	4102	78.5	1.319	0.888	0.989	0.938	13.05
J	4103	92.0	3.573	0.990	0.945	0.967	8.05
M	4101	89.0	29.543	0.948	-0.051	0.448	32.20
M	4102	85.5	30.324	0.938	0.194	0.566	28.74
M	4103	86.5	29.725	0.898	0.733	0.816	19.75
T	4101	58.5	1.904	0.160	0.746	0.453	37.81
T	4102	99.0	3.562	0.998	0.999	0.998	2.06
T	4103	97.5	3.708	0.999	0.972	0.985	5.24

The coordinate pattern in the second cohort is specific: M has $R^2_x \in [0.898, 0.948]$ and $R^2_y \in [-0.051, 0.733]$; all three C models have negative scores on both axes. This helps identify what those probes fail to recover, but it does not establish a universal ordering by probe quality. For example, first-cohort Z has $R^2_{\text{pos}} \in [0.641, 0.905]$ despite SR of 50.0–53.5%, while first-cohort G has lower mean probe scores than its second-cohort counterparts but similarly high SR.

F.3 Early alarms and final losses

The frozen offline alarm tests whether PR is below 2 at epoch 10. Figure 3c gives the main-paper view; Figure 4 includes all five first-cohort unguarded innovation runs and all five marginal runs measured by this audit. Among innovation runs, the two low-SR outcomes trigger, but so do all three high-SR outcomes. The counts are two true positives, zero false negatives, three false positives, and zero true negatives within that arm. None of the five marginal runs triggers. Distinguishing methods is therefore easier than predicting which innovation run will fail.

Table 26: **Final validation losses of all first-cohort unguarded innovation models.** Losses are taken from the final logged validation summary; SR is computed from the four planning evaluations.

Nominal label	SR (%)	Prediction loss	Regularizer loss
1001	98.0	0.007025	1.933798
1002	58.0	0.045080	2.705931
1003	97.5	0.005633	1.759733
1004	50.5	0.004808	2.631622
1005	99.5	0.003759	2.092502

Table 26 shows why low prediction loss is insufficient: the 50.5% planner has lower prediction loss than two models above 97% SR, while the other poor planner has the largest loss. The five-run Spearman correlation is -0.3, with a sample far too small to characterize a general relationship. In these records no one-sided prediction-loss threshold separates the two poor outcomes from all successful models. Final regularizer loss, in contrast, is highest for both poor planners; with five runs, we treat this as an observation rather than a diagnostic. The regularizer is also a soft finite-sample penalty; a low prediction loss does not establish that its Gaussian target, or state information, has been preserved.

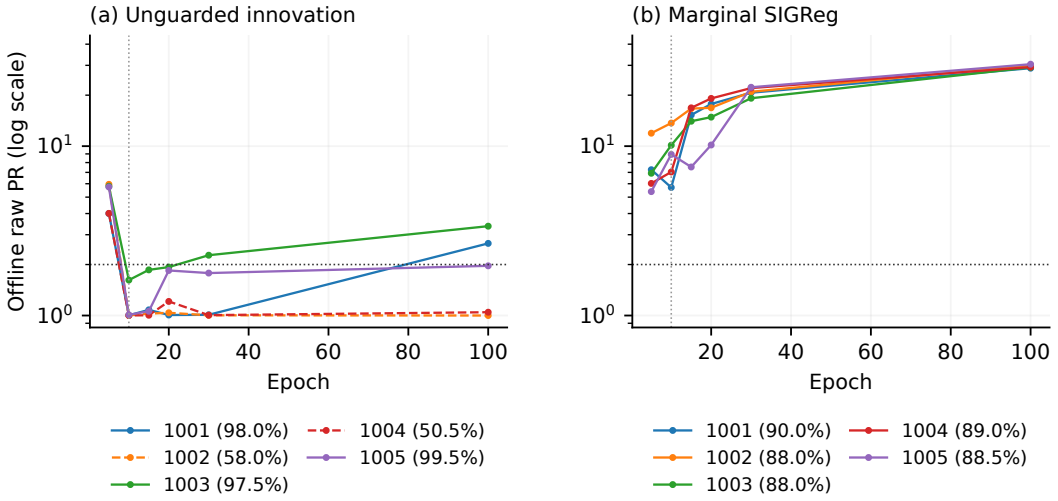

Figure 4: **Offline PR trajectories for every audited first-cohort innovation and marginal model.** Legend entries show nominal label and final planning SR. Dotted guides mark epoch 10 and PR 2; dashed model curves identify I labels 1002 and 1004, whose final SRs are 58.0% and 50.5%; no general SR cutoff is introduced. The vertical axis is logarithmic. All five innovation runs cross the assessment threshold despite different final outcomes. These offline values are not the smoothed online guard monitor.

G Controlled tests and limits of diagnostic explanations

G.1 A controlled persistence experiment

The controlled experiment holds the observation map, encoder, optimizer, and source-state distribution fixed while changing a scalar transition coefficient. Each sample is an independent two-state episode. For coordinate $j \in \{1, 2\}$, draw the source from $s_j \sim \frac{1}{2}\mathcal{N}(-2, 0.25) + \frac{1}{2}\mathcal{N}(2, 0.25)$, independently across coordinates. The next state and observed input are

$$s' = \rho s + \sqrt{1-\rho^2}\,\varepsilon, \qquad \varepsilon \sim \mathcal{N}(0, I_2), \quad \varepsilon \perp s, \qquad x = (s_1 + 0.3s_2^3,\ s_2). \tag{40}$$

I_2 is the two-dimensional identity matrix. The same invertible shear maps s' to x'. The coefficients are $\rho \in \{0.50, 0.89, 0.99\}$. The source distribution is fixed, but the next-state distribution changes with ρ; this is not an intervention holding both marginals constant.

The encoder is a $2 \to 64 \to 64 \to 2$ GELU MLP, with a linear predictor for the alignment term. All arms use 100,000 training pairs, 10,000 test pairs, batch size 256, 60 epochs, AdamW at 10^{-3}, and outer weight $\lambda = 0.01$. I and T share anchor weight 0.1 and detached EMA statistics with update weight 0.01. The three model seeds are 5101–5103; fixed data seeds 7001 and 7003 generate train and test pairs. No validation set is used. The linear state probe uses ridge coefficient 10^{-3}, fitted on training embeddings and scored on independent test pairs, without outcome-based tuning. Three additional marginal-SIGReg controls use stationary Gaussian source states at $\rho = 0.89$.

Table 27 retains each of the 30 individual outcomes and adds residual-dependence and variance measurements. The stationary Gaussian control and the absence of a monotone I–T advantage are discussed in Section 4.6. These observations do not exclude an effect of persistence in other systems.

Two earlier persistence summaries helped motivate this comparison but are not directly interchangeable. The approximately 0.89 Reacher value summarizes a true-angle transition estimate at stride 64. The 0.9355 Two-Room value is a median of learned-embedding autocorrelation summaries over 14 historical checkpoints. The offline Two-Room script uses adjacent frames from its eight-frame manifest, whereas training uses frameskip 5. The variables, sampling intervals, and aggregation differ; similar numerical values do not establish matched dynamical regimes.

Table 27: **Every controlled-system state-decoding outcome.** Columns 5101–5103 are model seeds. "Stat." denotes the Gaussian-source positive control. c is mean residual–state dependence as defined in the text; v is mean residual coordinate variance. The latter two columns average the per-model statistics over three runs.

ρ	Arm	5101	5102	5103	Mean R^2	c	v
0.50	M	0.953	0.948	0.951	0.951 ± 0.002	0.462	0.412
0.50	I	0.840	0.836	0.841	0.839 ± 0.003	0.605	0.603
0.50	T	0.980	0.982	0.491	0.817 ± 0.282	0.735	1.027
0.89	M	0.484	0.976	0.485	0.648 ± 0.284	0.365	0.931
0.89	I	0.946	0.825	0.483	0.751 ± 0.240	0.364	0.965
0.89	T	0.988	0.499	0.992	0.826 ± 0.284	0.461	0.986
0.99	M	0.825	0.823	0.469	0.706 ± 0.205	0.061	0.907
0.99	I	0.664	0.567	0.410	0.547 ± 0.128	0.193	1.652
0.99	T	0.414	0.762	0.412	0.529 ± 0.201	0.072	0.976
0.89	M (stat.)	0.989	0.990	0.988	0.989 ± 0.001	0.012	0.993

The next state is an equal mixture of $\mathcal{N}(\pm 2\rho,\, 1 - 0.75\rho^2)$, which is non-Gaussian for every $\rho > 0$; its measured excess kurtosis is -0.584, -1.564, and -1.755 at the three coefficients. The $\rho = 0.50$ marginal is thus closest to Gaussian, which may contribute to the marginal arm's advantage in that cell.

G.2 Estimator fidelity and residual dependence

The controlled harness uses a separate EMA implementation: its batch mean averages the two endpoint means, and its variance averages the two endpoint sample variances. The lag estimate is updated using embeddings standardized by the newly updated EMA moments. The temporal-difference scale is another running variance. These choices are shared within the controlled comparison but differ in detail from the production normalizers documented in the methodology. The experiment therefore tests the proposed persistence explanation in a fixed auxiliary system, not a bitwise copy of either benchmark's training path.

Its residual–state diagnostic is $c = \max_{i,j} |\operatorname{Corr}(r_i, s_j)|$, where r is the evaluated residual, s the source state, and $i, j \in \{1, 2\}$. For I and M the diagnostic uses the estimated autoregressive residual; for T it uses the standardized difference. A 1,000-permutation reference shuffles residual rows relative to states and supplies the 95th percentile of c, approximately 0.025. Cell-mean dependence in the mixture-source conditions ranges from 0.061 to 0.735, while the stationary control averages 0.012. These correlations show remaining linear dependence in the measured residuals. Passing this diagnostic would still not prove statistical independence.

The fixed-embedding diagnostics in Table 28 provide a complementary check without representation learning. In transient examples the estimated residual variance can differ from one despite near-zero excess kurtosis. The coefficient clamp also matters near unit persistence. These observations separate distributional shape, scale, estimator accuracy, and dependence; none is interchangeable with the complete independent-noise assumption.

The diagnostic uses identity embeddings of a known Gaussian-noise autoregression, with 400 fitting and 400 evaluation trajectories of 50 states per case. A single random-number generator with seed 42 generates the cases in a fixed order. The fitting trajectories supply a shared mean and standard deviation over all their states, followed by a diagonal lag estimate from standardized adjacent pairs, clipped to $[-0.995, 0.995]$. These fitted statistics remain fixed on the separate evaluation trajectories. This is a pooled-statistics diagnostic, not a reproduction of the evolving EMA in neural training.

At $\rho = 0.9$, stationary starts produce residual variance 0.992. The shifted case initializes states from $\mathcal{N}(2, 4)$; the bimodal case uses the equal mixture of $\mathcal{N}(-2, 0.09)$ and $\mathcal{N}(2, 0.09)$. The second argument denotes variance. Although the transition coefficient and Gaussian-noise law are unchanged, the estimated residual variances become 0.671 and 0.786, with excess kurtoses -0.010 and -0.012. These results distinguish an approximately Gaussian shape from the unit scale targeted by the regularizer. At $\rho = 0.999$, the clipped coefficient is 0.995 and residual variance is 0.203, illustrating an additional distortion near the clamp.

The shuffled control permutes all evaluation frames globally before forming adjacent pairs; its statistics are still fitted on intact fitting trajectories. Its variance 9.610, correlation 0.672, and excess kurtosis -0.028 therefore test broken temporal correspondence under an unchanged fitting procedure. The stationary and shuffled cases use separate random draws, not the same evaluation frames before and after shuffling. For this diagnostic, c and c_{abs} use only the first coordinate: $c = |\operatorname{Corr}(\eta_1, z_1)|$ and $c_{\mathrm{abs}} = |\operatorname{Corr}(|\eta_1|, |z_1|)|$, where z is the preceding standardized state. Small values of these first-coordinate summaries do not rule out cross-coordinate dependence or validate the diagonal transition model.

Table 28: **Fixed-embedding diagnostics.** Variance and excess kurtosis describe the estimated residual. c and c_{abs} are the first-coordinate residual–state correlation summaries before and after taking absolute values. These experiments use their own diagnostic estimator and do not establish full-vector or temporal independence.

Condition	Variance	Excess kurtosis	c	c_{abs}
stationary rho=0.0	0.997	-0.009	0.008	0.005
stationary rho=0.2	1.013	0.036	0.000	0.011
stationary rho=0.6	0.991	-0.006	0.009	0.000
stationary rho=0.9	0.992	-0.042	0.005	0.002
stationary rho=0.99	1.099	0.029	0.006	0.006
stationary rho=0.999	0.203	0.019	0.082	0.003
shifted_var4 rho=0.9	0.671	-0.010	0.008	0.010
bimodal rho=0.9	0.786	-0.012	0.005	0.000
aniso axis-aligned	1.010	0.008	0.001	0.004
aniso rotated45	0.995	0.026	0.006	0.009
shuffled pairs rho=0.9	9.610	-0.028	0.672	0.414

G.3 The final-checkpoint innovation statistic as a diagnostic

The diagnostic registry contains 299 final-checkpoint records across development settings. The analysis recomputes a common innovation statistic on a fixed sample of up to 4,096 pairs from each run's own training-pair file, using projection seed 12345 and pooled endpoint normalization. These pairs come from the training distribution and may overlap with encoder training data, so the diagnostic is post hoc rather than a held-out evaluation.

Across these records the Spearman correlation between the innovation statistic and angle-decoding R^2 is 0.034. Table 29 gives the subgroup correlations. The trajectory-only correlations are nonnegative in this registry, ranging from approximately zero to 0.496. Because lower statistic values are preferable for the training target, these weak or positive associations also fail to establish a consistently useful quality ranking. The two OU subgroups have negative associations. This mixture of settings makes the pooled null descriptive rather than a universal claim about Gaussianity metrics.

Table 29: **Post-hoc diagnostic associations by data setting.** r_s is Spearman rank correlation between the innovation statistic and joint-angle R^2. n counts archived records, not independent datasets.

Data	Stride / coefficient	n	r_s
OU	0.95	12	-0.154
OU	0.99	12	-0.643
Trajectory	1	36	0.496
Trajectory	2	36	0.254
Trajectory	8	48	0.496
Trajectory	16	82	0.100
Trajectory	64	73	0.000

H Provenance, reproduction, and evaluation comparability

The accompanying result package preserves the primary CSV records, geometry summaries, per-evaluation historical scores extracted from archived logs, and a script that regenerates the displayed

numerical tables and vector figures. It preserves the two main planning cohorts, the 50-run Reacher cohort, the nine-run statistics comparison, and the 30-run controlled experiment as separate datasets. The main planning means were checked against the logged episode-success arrays: 224 evaluations across the two cohorts, each containing 50 episodes. An additional 168 distinct historical evaluations were checked by the same procedure; four also appear in a second historical table context. These checks verify score aggregation, not retraining or checkpoint reevaluation.

The anonymized supplementary material contains the training code, the primary result records, and a script that regenerates every table and figure. Checkpoints are omitted because of upload limits; full training checkpoints store the normalizer, guard, optimizer, scheduler, and random-generator states needed to resume training.

The stored execution record specifies PyTorch 2.9.0 with CUDA 13.0 for the Reacher stack and PyTorch 2.10.0 with CUDA 12.8 for the LeWM stack. The main planning cohorts trained on an RTX 5090 and evaluated on an RTX 3060. Recorded profiling for a regenerated-data Two-Room model was approximately 2.75 training GPU-hours; this is a configuration-specific measurement, not an efficiency comparison between regularizers. We do not report a new training reproduction or an audited total compute budget for the entire development program.

An evaluation discrepancy motivated comparisons within one recorded harness. A published PushT reference checkpoint scores 90.0, 80.0, 84.0, and 96.0% under evaluation seeds 0, 1, 2, and 42, respectively, giving $87.5 \pm 7.0\%$ across evaluations. The locally trained marginal model scores $86.0 \pm 2.3\%$. Because the reference checkpoint's score in our harness differs from its published value, we compare methods only within this harness.

I Additional related work and distinctions

Learning representations for prediction and control. Contrastive Predictive Coding learns representations through prediction of future latent features with a contrastive objective (van den Oord et al., 2019). I-JEPA learns by predicting representations of image regions (Assran et al., 2023). DINO-WM uses pretrained visual features as the state space for learned dynamics and goal-directed planning (Zhou et al., 2025). V-JEPA 2 combines video representation learning with action-conditioned post-training for robotic planning (Assran et al., 2025). LeWorldModel instead studies joint training of the representation and predictor from pixels with prediction and Gaussian regularization (Maes et al., 2026). This last setting exposes the regularizer's influence on the coordinates used by the dynamics model. Our intervention concerns that training objective. We do not evaluate whether the same intervention improves a frozen pretrained encoder or a large video model.

Gaussian targets and temporal residuals. Table 30 compares the objects constrained by the closest Gaussian-regularization approaches. This comparison locates the proposed construction without asserting that temporal residual regularization itself is new. In particular, TC-LeWM already applies SIGReg to a temporal residual; our distinction is the estimated autoregressive subtraction and its normalization, accompanied by an explicit study of statistics gradients. The version of TC-LeWM used for this comparison employs four-frame windows and evaluates a policy trained on frozen learned features (Liu et al., 2026). Our adapted regularizer is evaluated through LeWorldModel planning. Its score in that setting does not establish a ranking of the original robot-learning systems.

Recovery from temporal observations. Under its nonlinear autoregressive model and variability conditions, Independent Innovation Analysis recovers innovation components up to reordering and a separate invertible transformation of each component (Morioka et al., 2021). Such transformations can be nonlinear, so this recovery notion differs from decoding state variables by linear regression in our experiments. LEAP and temporally disentangled representation learning likewise exploit assumptions about temporal generative structure to recover latent processes or components (Yao et al., 2022b;a). Innovation SIGReg imposes neither their generative objectives nor their complete identification conditions. The oracle calculation in Appendix A motivates a target distribution; it does not show that optimizing our finite-sample loss recovers either the generating states or their independent noise.

Table 30: **Objects receiving Gaussian regularization.** The table describes the regularizer, not an equivalence of architectures, training protocols, or evaluation tasks. The two simple controls are defined fully in Appendix B.

Approach	Regularized object	Relation to Innovation SIGReg
Marginal SIGReg (Balestriero & LeCun, 2025)	Raw embeddings.	Retained as a weighted anchor.
Sub-JEPA (Zhao et al., 2026)	Embeddings projected into random subspaces.	Alters spatial scope without estimating a transition residual.
TC-LeWM (Liu et al., 2026)	Deviations from a local temporal mean.	Centers a temporal window without estimating an autoregressive coefficient or using the AR(1) residual scale.
Normalization control	Standardized next embeddings, plus a raw anchor.	Tests normalization without temporal subtraction.
Difference control	Standardized temporal differences, plus a raw anchor.	Tests a simpler temporal target without estimating persistence.
Innovation SIGReg	Normalized persistence-adjusted residuals, plus a raw anchor.	Separates residual construction from statistics freshness and differentiability.

Stopping gradients. Stop-gradient is an established ingredient in self-supervised learning, including BYOL and SimSiam (Grill et al., 2020; Chen & He, 2020). BYOL also updates a target network through an exponential moving average of online-network parameters. Normalization provides a different precedent: Batch Renormalization stops gradients through batch-to-running correction factors while retaining derivatives through batch normalization (Ioffe, 2017). Our intervention instead treats the estimated mean, variance, and lag coefficient as constants during backpropagation, retaining direct gradients through both residual endpoints. Its running averages track statistics, not target-network parameters. The contribution is the measured, benchmark-dependent effect of this gradient choice within the transition regularizer.

J Extended discussion and interpretation

J.1 What the mathematical motivation establishes

An invertible affine map preserves non-Gaussianity. Exact Gaussian marginal matching can therefore conflict with preserving affine state coordinates. A finite penalty, however, need not destroy information, and useful control need not require recovery of every physical variable.

The oracle example isolates one reason why a transition target can be appropriate. Under the specified linear–Gaussian dynamics, the true transition noise has a fixed Gaussian law even when episodic pooling changes the state distribution. The implemented residual remains an approximation. Shared endpoint normalization, coefficient estimation, clipping, and a coordinate system that mixes distinct transition rates can all separate it from that oracle (Section A.3). Moreover, the raw-marginal anchor retains a competing preference over embedding shape. The method rebalances these pressures; it does not eliminate marginal regularization or solve the general mismatch exactly.

J.2 What the gradient intervention establishes

At fixed embeddings and random projections, the current-batch modes have identical statistics, residuals, and losses. Only one includes derivatives through the statistics. Their subsequent parameters and losses can therefore diverge; equality of one forward computation does not establish equivalent optimization or predict which mode will be better.

The Two-Room outcome is consistent with those derivative paths being consequential for optimization. It does not isolate a complete failure mechanism: initial parameter and random-number matching remain unverified, and three models per configuration cannot establish a general failure rate. Reacher supplies a further boundary. Current-batch detachment performs worse than both running-statistics detachment and current-batch differentiation in its statistics comparison. Reduced estimator noise or slower changes in normalization are possible explanations for the running estimates' benefit, but the

available experiments do not isolate either explanation from sampling, architecture, dimension, or objective differences.

J.3 Representation measurements and planning

Dimensional collapse, in which embeddings occupy a lower-dimensional subspace, motivates inspecting representation spectra (Jing et al., 2022). RankMe uses an entropy-based effective rank of feature singular values to assess downstream representation quality (Garrido et al., 2023). Our participation ratio instead measures the spread of covariance eigenvalues; the two statistics are distinct. In our planning experiments, low participation ratio can accompany strong performance, so variance concentration alone does not establish a loss of task-relevant information. Likewise, position probes evaluate encoded observations, not action-conditioned prediction along candidate plans. The similar mean probe scores of the two batch-statistics variants are descriptive, not evidence of statistical equivalence.

The exploratory Two-Room probes split episodes between probe fitting, validation, and testing, but do not establish that the encoder was trained on disjoint episodes. The inherited Reacher probe evaluates a different angle distribution from the one used for fitting. These protocols answer specific accessibility questions. They do not establish generalization to arbitrary states, an information ceiling, or sufficient conditions for successful control (Appendices D and F).

J.4 Transfer and remaining empirical questions

The strongest batch-detached configuration has only been evaluated in the regenerated Two-Room setting. Official-data Two-Room and PushT results concern an earlier unguarded configuration, so they cannot resolve its transfer. The guard's repeated means provide useful evidence for an alternative intervention, but the fixed and scheduled anchors prevent attributing its value specifically to adaptive triggering. The controlled autoregressive study also leaves the choice of residual unresolved: changing persistence changes the next-state distribution, and the relative decoding results are nonmonotone.

Larger comparisons with recorded initialization and random-number states can test the gradient intervention more sharply. Official-data and manipulation evaluations can test transfer. Joint analysis of encoded states, predicted rollouts, and planner costs could then clarify why accessible state information does not always translate into successful control.