WHEN DYNAMICS ARE SUBOPTIMAL: ISOLATING DYNAMICS INDUCTIVE BIAS IN REPRESENTATION LEARNING

Anonymous authors
Paper under double-blind review

ABSTRACT

In many dynamical models, representations and dynamics are learned jointly, and the chosen dynamics model class and fitting objective thereby impose an inductive bias on representation learning. At a given training iterate, the dynamics parameters can be suboptimal for the representation under the chosen fitting criterion, so the induced representation gradient can differ from the corresponding gradient under a fitted dynamics reference. We theoretically characterize this representation-gradient discrepancy. In the canonical case where the dynamics-induced representation gradient is obtained by differentiating the dynamics-fitting objective with respect to the representation parameters, the discrepancy is exactly the gradient of the dynamics-parameter suboptimality gap. Controlled experiments in a sequential latent-variable model show that dynamics-parameter suboptimality alters representation formation, with consequences for learned dynamical quality. The effect is most pronounced during early representation formation, motivating Asymmetric Coupling Warm-up (ACW), a lightweight intervention that temporarily attenuates dynamics-to-representation coupling early in training. Applying ACW across distinct representation–dynamics paradigms yields consistent changes in optimization trajectories and downstream outcomes, providing cross-paradigm evidence that the identified mechanism has observable consequences beyond the controlled setting.

1 INTRODUCTION

For dynamical systems observed through complex measurements, a broad class of approaches jointly learns a latent representation and the dynamics governing its evolution in latent space. Such approaches span probabilistic latent state-space models, world models, self-predictive learning, Koopman representation learning, and reduced-order modeling (Karl et al., 2017; Hafner et al., 2019; Lusch et al., 2018; Khetarpal et al., 2025; Friedl et al., 2025). The inductive role of dynamics is particularly explicit in Koopman representation learning, where the representation is learned under an objective that encourages its latent evolution to be captured by a linear operator (Lusch et al., 2018). More generally, joint representation–dynamics learning is bidirectional: the dynamics model is fitted to the learned representation, while the chosen dynamics model class and fitting objective together impose an inductive bias on representation learning (Cheng et al., 2026).

Under standard joint training, however, the representation and dynamics parameters are optimized jointly, so the current dynamics parameters need not yet be well fitted to the current representation. Prior work in self-predictive learning has shown that the extent to which a predictor is optimized for the current representation can materially affect representation learning (Tang et al., 2023). Our focus is more specific: we examine how the dynamics-induced representation gradient under the current dynamics parameters differs from the corresponding gradient under a dynamics reference fitted to that same representation. When the current dynamics parameters are suboptimal, does this discrepancy materially alter representation learning?

This discrepancy has a direct mechanistic interpretation. The gradient induced by the fitted dynamics reference captures how the chosen dynamics model class and fitting objective shape the representation, whereas the gradient discrepancy reflects the additional dependence on the current dynamics

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.

parameters. In the canonical case, the dynamics-induced representation gradient is obtained by differentiating the dynamics-fitting objective with respect to the representation parameters. Under this condition, the distinction becomes exact: the objective decomposes into an optimal value function $A_{\mathcal{P}}$ and a corresponding suboptimality gap $O_{\mathcal{P}}$, and the representation gradient of $O_{\mathcal{P}}$ is exactly the gradient discrepancy above. In this sense, the dynamics inductive bias is separated from the additional influence introduced by dynamics-parameter suboptimality.

We conduct a controlled study in a sequential latent-variable model and show that dynamics-parameter suboptimality materially alters the course of representation formation and the dynamical quality of the learned representation. Within this controlled setting, the influence is most pronounced early in representation formation, motivating Asymmetric Coupling Warm-up (ACW), a lightweight intervention that temporarily attenuates dynamics-to-representation coupling during early training. We then extend the analysis across three substantially different paradigms—Koopman-AE, PLDM, and STORM—spanning linear latent dynamics, deterministic self-prediction, and stochastic world modeling, and compare ACW with each model's original training procedure. The resulting differences in optimization trajectories and downstream outcomes show that dynamics-to-representation coupling has measurable consequences across substantially different representation–dynamics paradigms. Our contributions are summarized as follows:

- **Theoretical characterization.** We formalize the distinction between dynamics inductive bias and the additional influence on representation learning arising from dynamics-parameter suboptimality.
- **Controlled mechanism diagnosis.** In a sequential latent-variable model, we show that dynamics-parameter suboptimality alters representation formation, with consequences for learned dynamical quality.
- **Cross-paradigm evidence.** We introduce the lightweight coupling intervention ACW and provide evidence that the identified dynamics-parameter suboptimality mechanism has observable consequences across distinct representation–dynamics paradigms.

2 Related Works

2.1 Representation–Dynamics Learning Regimes

A line of work learns representations with respect to optimally fitted dynamics models. Koopman-based methods use least-squares-fitted linear operators (Takeishi et al., 2017), while recent self-predictive and evolution-operator formulations characterize optimal predictors for fixed representations (Khetarpal et al., 2025; Turri et al., 2026). These approaches exploit structure in the inner dynamics-fitting problem, allowing the fitted predictor to be obtained analytically or characterized explicitly.

Other approaches decouple representation and dynamics optimization through frozen or staged training. DINO-WM learns dynamics over frozen pretrained visual features (Zhou et al., 2025), while staged world-modeling approaches separately train representations and dynamics (Ha & Schmidhuber, 2018) or pretrain an action-free predictive model before action-conditioned world-model finetuning (Seo et al., 2022). Together, these works illustrate distinct regimes for organizing representation and dynamics learning. Joint learning is particularly relevant to our setting because it preserves dynamics-induced representation shaping while exposing this interaction throughout optimization.

2.2 Regulating Representation–Dynamics Coupling

Prior work regulates representation–dynamics coupling through target asymmetry and gradient routing, predictor optimization timescales, and staged training schedules. SPR uses stop-gradient EMA targets (Schwarzer et al., 2021), DreamerV2/V3 and STORM employ asymmetric gradient routing (Hafner et al., 2021; 2025; Zhang et al., 2023), Tang et al. (2023) study semi-gradient learning with faster or optimal predictors, and CoLA-World introduces warm-up before joint co-evolution (Wang et al., 2026). These approaches regulate how and when dynamics influences representation learning, with several also highlighting the importance of predictor quality or optimization state. Our work examines the mechanism underlying this influence, distinguishing the intended dynamics inductive bias from the additional effect of dynamics-parameter suboptimality.

3 DYNAMICS FITTING AND REPRESENTATION-GRADIENT DISCREPANCY

Consider a representation model with parameters $\boldsymbol{\phi} \in \Phi \subseteq \mathbb{R}^{d_\phi}$ jointly learned with a dynamics model parameterized by $\boldsymbol{\psi} \in \Psi \subseteq \mathbb{R}^{d_\psi}$. Let $E_{\text{fit}}(\boldsymbol{\phi}, \boldsymbol{\psi})$ denote the objective used to fit the dynamics model, and let $\mathbf{g}_{\boldsymbol{\phi}}^{\text{dyn}}(\boldsymbol{\phi}, \boldsymbol{\psi})$ denote the dynamics-induced contribution to the representation gradient, evaluated at the current dynamics parameters $\boldsymbol{\psi}$ under the model's native gradient routing. Other model quantities and the fitting setup are omitted from the notation and kept unchanged in the comparison below.

For a fixed $\boldsymbol{\phi}$ at which the dynamics-fitting minimum is attained, let

$$\boldsymbol{\psi}_{\text{fit}}^{\star}(\boldsymbol{\phi}) \in \arg\min_{\boldsymbol{\psi} \in \Psi} E_{\text{fit}}(\boldsymbol{\phi}, \boldsymbol{\psi})$$

denote a selected fitted optimum. We define the representation-gradient discrepancy relative to this fitted reference as

$$\Delta \mathbf{g}_{\boldsymbol{\phi}} := \mathbf{g}_{\boldsymbol{\phi}}^{\text{dyn}}(\boldsymbol{\phi}, \boldsymbol{\psi}) - \mathbf{g}_{\boldsymbol{\phi}}^{\text{dyn}}(\boldsymbol{\phi}, \boldsymbol{\psi}_{\text{fit}}^{\star}(\boldsymbol{\phi}))\,. \tag{1}$$

3.1 REPRESENTATION GRADIENTS UNDER DYNAMICS-PARAMETER SUBOPTIMALITY

We consider the common setting in which the dynamics-fitting objective also induces the dynamics-related gradient to the representation, i.e., $\mathbf{g}_{\boldsymbol{\phi}}^{\text{dyn}} = \nabla_{\boldsymbol{\phi}} E_{\text{fit}}$. For brevity, write $E := E_{\text{fit}}$, and let $\mathcal{P} = \{T_{\boldsymbol{\psi}} : \boldsymbol{\psi} \in \Psi\}$ denote the chosen dynamics model class. We define the optimal value function and the corresponding suboptimality gap as

$$A_{\mathcal{P}}(\boldsymbol{\phi}) := \inf_{\boldsymbol{\psi} \in \Psi} E(\boldsymbol{\phi}, \boldsymbol{\psi}), \qquad O_{\mathcal{P}}(\boldsymbol{\phi}, \boldsymbol{\psi}) := E(\boldsymbol{\phi}, \boldsymbol{\psi}) - A_{\mathcal{P}}(\boldsymbol{\phi}). \tag{2}$$

From these definitions, $O_{\mathcal{P}}(\boldsymbol{\phi}, \boldsymbol{\psi}) \geq 0$ and

$$E(\boldsymbol{\phi}, \boldsymbol{\psi}) = A_{\mathcal{P}}(\boldsymbol{\phi}) + O_{\mathcal{P}}(\boldsymbol{\phi}, \boldsymbol{\psi}).$$

For a fixed representation, $A_{\mathcal{P}}$ is independent of the current dynamics parameters, whereas $O_{\mathcal{P}}$ quantifies the gap between the current fitting value and this optimum. When the terms are differentiable with respect to $\boldsymbol{\phi}$, the definition $E = A_{\mathcal{P}} + O_{\mathcal{P}}$ immediately gives

$$\nabla_{\boldsymbol{\phi}} E = \nabla_{\boldsymbol{\phi}} A_{\mathcal{P}} + \nabla_{\boldsymbol{\phi}} O_{\mathcal{P}}.$$

The decomposition itself is therefore definitional. The key step for connecting it to the fitted-dynamics reference in Equation 1 is to characterize the gradient of the optimal-value function $A_{\mathcal{P}}$.

Proposition 1 (Gradient of the Fitted-Dynamics Value Function)

Suppose $\boldsymbol{\phi} \in \operatorname{int} \Phi$, the dynamics-fitting minimum is attained, the directional form of Danskin's theorem (Danskin, 1967) applies at $\boldsymbol{\phi}$, and $A_{\mathcal{P}}$ is differentiable at $\boldsymbol{\phi}$. Then, for any fitted optimum

$$\boldsymbol{\psi}_{\text{fit}}^{\star}(\boldsymbol{\phi}) \in \arg\min_{\boldsymbol{\psi} \in \Psi} E(\boldsymbol{\phi}, \boldsymbol{\psi}),$$

we have

$$\nabla_{\boldsymbol{\phi}} A_{\mathcal{P}}(\boldsymbol{\phi}) = \nabla_{\boldsymbol{\phi}} E(\boldsymbol{\phi}, \widetilde{\boldsymbol{\psi}})\Big|_{\widetilde{\boldsymbol{\psi}} = \boldsymbol{\psi}_{\text{fit}}^{\star}(\boldsymbol{\phi})}\,. \tag{3}$$

Here, $\widetilde{\boldsymbol{\psi}}$ is treated as constant when differentiating with respect to $\boldsymbol{\phi}$; in particular, no derivative is taken through the mapping $\boldsymbol{\psi}_{\text{fit}}^{\star}(\boldsymbol{\phi})$. The proof is provided in Appendix A.1.

Consequently, using $O_{\mathcal{P}} = E - A_{\mathcal{P}}$, *Proposition* 1 gives

$$\nabla_{\boldsymbol{\phi}} O_{\mathcal{P}}(\boldsymbol{\phi}, \boldsymbol{\psi}) = \nabla_{\boldsymbol{\phi}} E(\boldsymbol{\phi}, \boldsymbol{\psi}) - \nabla_{\boldsymbol{\phi}} E(\boldsymbol{\phi}, \widetilde{\boldsymbol{\psi}})\Big|_{\widetilde{\boldsymbol{\psi}} = \boldsymbol{\psi}_{\text{fit}}^{\star}(\boldsymbol{\phi})}\,. \tag{4}$$

Thus, in the setting considered here where $\mathbf{g}_{\boldsymbol{\phi}}^{\text{dyn}} = \nabla_{\boldsymbol{\phi}} E$, combining equation 4 with equation 1 yields

$$\Delta \mathbf{g}_{\boldsymbol{\phi}} = \nabla_{\boldsymbol{\phi}} O_{\mathcal{P}}(\boldsymbol{\phi}, \boldsymbol{\psi}).$$

Hence, the representation-gradient discrepancy between the current and fitted dynamics parameters is exactly the representation gradient of the dynamics-parameter suboptimality gap.

4 A CONTROLLED STUDY OF DYNAMICS-PARAMETER SUBOPTIMALITY

In this section, we use controlled experiments to examine how dynamics-parameter suboptimality affects joint representation learning, focusing on its effects on representation updates, optimization trajectories, and the resulting representation–dynamics learning outcomes. A sequential VAE provides an exact probabilistic instantiation of this setting: its dynamics-fitting objective both trains the generative transition model and induces a dynamics-related gradient to the representation parameters.

Sequential latent-variable models are commonly trained by amortized variational inference with a sequential evidence lower bound (ELBO) (Bayer et al., 2021), whose negative form can be expressed in terms of distortion and rate (Klushyn et al., 2021). Following the notation above, $\boldsymbol{\phi}$ denotes the representation parameters, instantiated here as the parameters of the amortized inference model, $\boldsymbol{\psi}$ denotes the latent dynamics parameters, and $\boldsymbol{\theta}$ parameterizes the observation model. Let $X = X_{1:T}$ and $Z = Z_{1:T}$ denote the observation and latent random sequences, with realizations $\mathbf{x}$ and $\mathbf{z}$. We write $q_{\boldsymbol{\phi}}(\mathbf{z} \mid \mathbf{x})$, $p_{\boldsymbol{\psi}}(\mathbf{z})$, $p_{\boldsymbol{\theta}}(\mathbf{x} \mid \mathbf{z})$, and $p_{\mathrm{data}}(\mathbf{x})$ for the amortized posterior, sequential latent prior, observation model, and data distribution, respectively. We consider the β-weighted negative ELBO

$$\mathcal{L}_\beta(\boldsymbol{\theta}, \boldsymbol{\phi}, \boldsymbol{\psi}) = D(\boldsymbol{\theta}, \boldsymbol{\phi}) + \beta R(\boldsymbol{\phi}, \boldsymbol{\psi}),$$

with

$$D(\boldsymbol{\theta}, \boldsymbol{\phi}) := -\mathbb{E}_{\mathbf{x}\sim p_{\mathrm{data}}} \mathbb{E}_{\mathbf{z}\sim q_{\boldsymbol{\phi}}(\mathbf{z}|\mathbf{x})} \left[\log p_{\boldsymbol{\theta}}(\mathbf{x} \mid \mathbf{z})\right],$$

and

$$R(\boldsymbol{\phi}, \boldsymbol{\psi}) := \mathbb{E}_{\mathbf{x}\sim p_{\mathrm{data}}} \left[D_{\mathrm{KL}}\left(q_{\boldsymbol{\phi}}(\mathbf{z} \mid \mathbf{x}) \,\|\, p_{\boldsymbol{\psi}}(\mathbf{z})\right)\right].$$

Here, D characterizes the reconstruction cost, while R measures the discrepancy between the amortized posterior and the generative latent model; $\beta > 0$ controls their relative weighting. Using the temporal factorizations

$$q_{\boldsymbol{\phi}}(\mathbf{z} \mid \mathbf{x}) = \prod_{t=1}^{T} q_{\boldsymbol{\phi}}(z_t \mid z_{<t}, \mathbf{x}), \qquad p_{\boldsymbol{\psi}}(\mathbf{z}) = \prod_{t=1}^{T} p_{\boldsymbol{\psi}}(z_t \mid z_{<t}),$$

we obtain the following exact decomposition. The full derivation and proof are provided in Appendix A.2.

Proposition 2 (Rate Decomposition in Sequential Latent-Variable Models)

Define the data–posterior joint distribution $r_{\boldsymbol{\phi}}(\boldsymbol{x}, \boldsymbol{z}) := p_{\mathrm{data}}(\boldsymbol{x}) q_{\boldsymbol{\phi}}(\boldsymbol{z} \mid \boldsymbol{x})$. *Then*

$$R(\boldsymbol{\phi}, \boldsymbol{\psi}) = C(\boldsymbol{\phi}) + E(\boldsymbol{\phi}, \boldsymbol{\psi}),$$

where

$$C(\boldsymbol{\phi}) := \sum_{t=1}^{T} I_{r_{\boldsymbol{\phi}}}\left(Z_t; X \mid Z_{<t}\right),$$

and

$$E(\boldsymbol{\phi}, \boldsymbol{\psi}) := \sum_{t=1}^{T} \mathbb{E}_{\boldsymbol{z}_{<t}\sim r_{\boldsymbol{\phi}}(\boldsymbol{z}_{<t})} \left[D_{\mathrm{KL}}\left(r_{\boldsymbol{\phi}}(\boldsymbol{z}_t \mid \boldsymbol{z}_{<t}) \| p_{\boldsymbol{\psi}}(\boldsymbol{z}_t \mid \boldsymbol{z}_{<t})\right)\right].$$

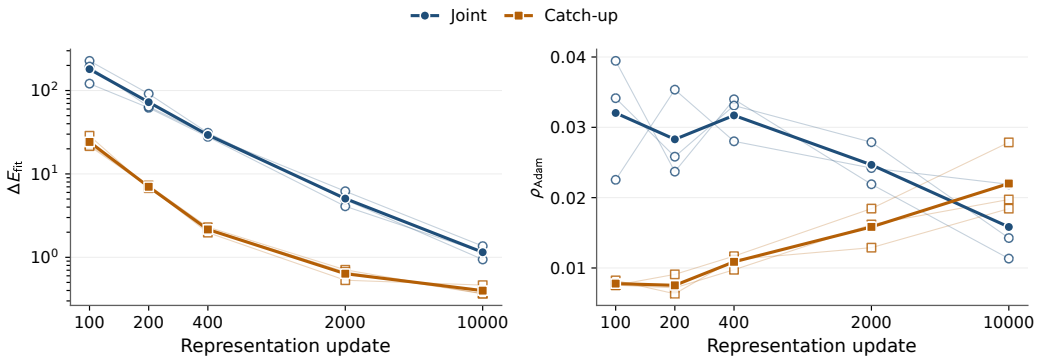
Here, C quantifies the dependence between the current latent state and the observations that remains after conditioning on the latent history, i.e., posterior information beyond latent history. The conditional $r_{\boldsymbol{\phi}}(z_t \mid z_{<t})$, induced by the data–posterior joint $r_{\boldsymbol{\phi}}$, defines the data-aggregated posterior transition, while E measures its discrepancy from the generative latent transition $p_{\boldsymbol{\psi}}(z_t \mid z_{<t})$ (with $t = 1$ corresponding to the initial prior). In this probabilistic setting, E is precisely the dynamics-fitting term, and the corresponding dynamics-induced contribution to the representation gradient is $\nabla_{\boldsymbol{\phi}} E$; in the full negative ELBO, this contribution is weighted by β.

The optimal-value decomposition of E therefore yields

$$R(\boldsymbol{\phi}, \boldsymbol{\psi}) = C(\boldsymbol{\phi}) + A_{\mathcal{P}}(\boldsymbol{\phi}) + O_{\mathcal{P}}(\boldsymbol{\phi}, \boldsymbol{\psi}). \tag{5}$$

Accordingly, the representation gradient of the negative ELBO becomes

$$\nabla_{\boldsymbol{\phi}} \mathcal{L}_\beta = \nabla_{\boldsymbol{\phi}} D + \beta \left(\nabla_{\boldsymbol{\phi}} C + \nabla_{\boldsymbol{\phi}} A_{\mathcal{P}} + \nabla_{\boldsymbol{\phi}} O_{\mathcal{P}}\right).$$



(a) (b)

Figure 1: Further transition fitting changes representation updates (a) Transition-fitting gain ΔE_{fit}. (b) Adam-level representation-update discrepancy ρ_{Adam}. Individual markers denote random seeds, and solid lines denote their mean. Both vertical axes are shown on a logarithmic scale.

The three rate terms correspond to posterior information beyond latent history, the dynamics-fitting optimal value, and dynamics-parameter suboptimality, respectively. Since D, C, and $A_{\mathcal{P}}$ are independent of the current dynamics parameters, their contributions cancel when comparing the representation gradients evaluated at the current dynamics parameters $\boldsymbol{\psi}$ and the fitted dynamics parameters $\boldsymbol{\psi}^{\star}_{\text{fit}}(\boldsymbol{\phi})$. *Proposition* 1 therefore gives

$$\nabla_{\boldsymbol{\phi}}\mathcal{L}_{\beta}(\boldsymbol{\theta},\boldsymbol{\phi},\boldsymbol{\psi}) - \nabla_{\boldsymbol{\phi}}\mathcal{L}_{\beta}(\boldsymbol{\theta},\boldsymbol{\phi},\widetilde{\boldsymbol{\psi}})\Big|_{\widetilde{\boldsymbol{\psi}}=\boldsymbol{\psi}^{\star}_{\text{fit}}(\boldsymbol{\phi})} = \beta\nabla_{\boldsymbol{\phi}}O_{\mathcal{P}}(\boldsymbol{\phi},\boldsymbol{\psi}). \tag{6}$$

Equation 6 motivates a practical reference-based diagnostic, where $\boldsymbol{\psi}^{\star}_{\text{fit}}(\boldsymbol{\phi})$ is approximated by finite dynamics-only optimization at fixed $\boldsymbol{\phi}$.

4.1 Dynamics-Parameter Suboptimality Alters Representation Updates

We study the mechanism on a visual double-pendulum system using a neural sequential VAE. For each of three random seeds, we conduct matched Joint and Catch-up runs with otherwise identical settings. Catch-up appends transition-only optimization between successive joint updates while holding the inference and observation models fixed; the number of extra updates is determined by a stabilization-based stopping criterion (Appendix B). To determine whether residual transition suboptimality affects representation learning, we retrospectively analyze selected checkpoints from the resulting training trajectories. At each checkpoint, let $\boldsymbol{\psi}_{\text{cur}}$ denote the saved transition parameters. Starting from $\boldsymbol{\psi}_{\text{cur}}$ and its saved optimizer state, we run an extended transition-only optimization to obtain a better-fitted reference $\boldsymbol{\psi}_{\text{ref}}$. We measure the resulting transition-fitting gain as

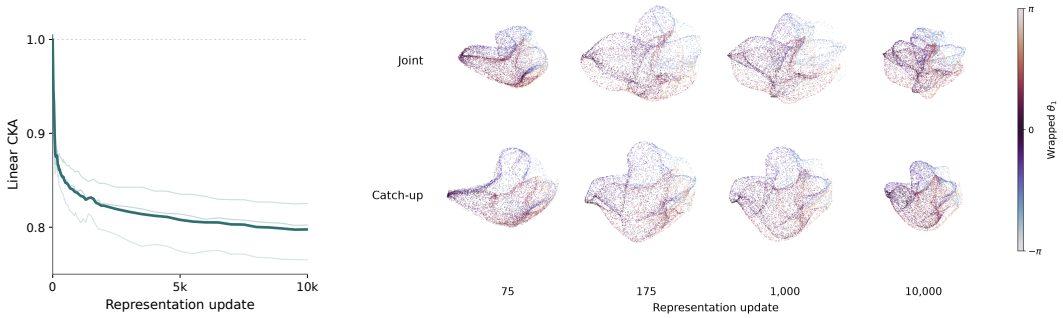
$$\Delta E_{\text{fit}} = E(\boldsymbol{\phi},\boldsymbol{\psi}_{\text{cur}}) - E(\boldsymbol{\phi},\boldsymbol{\psi}_{\text{ref}}).$$

Using the same inference-model Adam state, we then compute the representation updates at $\boldsymbol{\psi}_{\text{cur}}$ and $\boldsymbol{\psi}_{\text{ref}}$ with the complete training objective, so that the comparison reflects the optimizer update used in training under a fixed optimizer state. We define

$$\rho_{\text{Adam}} = \frac{\|\Delta\boldsymbol{\phi}_{\text{cur}} - \Delta\boldsymbol{\phi}_{\text{ref}}\|}{\max\left(\|\Delta\boldsymbol{\phi}_{\text{ref}}\|, \epsilon\right)},$$

where ϵ is a small constant for numerical stability. Since $\boldsymbol{\psi}_{\text{ref}}$ is obtained under a finite optimization budget, both quantities are reference-based diagnostics rather than exact evaluations at $\boldsymbol{\psi}^{\star}_{\text{fit}}(\boldsymbol{\phi})$.

Joint training exhibits substantial residual transition underfitting, most prominently during early representation formation. The transition-fitting gain ΔE_{fit} is large at early checkpoints and decreases as training proceeds (Figure 1a). At the same early checkpoints, ρ_{Adam} remains between approximately 2% and 4% across seeds (Figure 1b), showing that the current transition parameters measurably alter the actual representation update. Catch-up substantially suppresses both quantities during early representation formation (Figure 1). At later checkpoints, Catch-up can exhibit larger ρ_{Adam}; however, the two methods are no longer evaluated at a common representation, so this ordering no longer provides a controlled comparison of transition-parameter effects. We next ask whether these early update differences accumulate into distinct representation trajectories.



(a) Representation Similarity

(b) Latent Geometry

Figure 2: Dynamics-parameter suboptimality alters representation formation (a) Linear CKA between paired Joint and Catch-up representations (thin: seeds; thick: mean). (b) Latent-space evolution under shared PCA coordinates and final-scale alignment; color denotes wrapped θ_1.

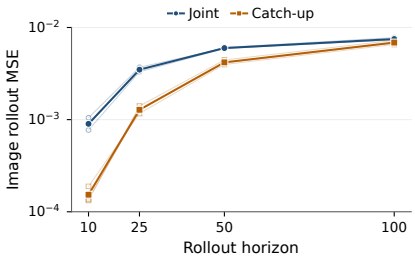


Figure 3: Multi-step image rollout under Joint and Catch-up training.

Table 1: Multi-step rollout with matched transition refits. Values are mean ± s.d. across three representation seeds, averaging two paired refit restarts per seed.

Horizon	Image MSE ($\times 10^{-3}$) ↓		Latent NMSE ↓	
	Joint	Catch-up	Joint	Catch-up
10	0.256 ± 0.028	$\mathbf{0.240} \pm 0.044$	0.023 ± 0.002	$\mathbf{0.021} \pm 0.006$
25	1.996 ± 0.027	$\mathbf{1.664} \pm 0.176$	0.256 ± 0.009	$\mathbf{0.186} \pm 0.019$
50	5.003 ± 0.250	$\mathbf{4.553} \pm 0.180$	0.916 ± 0.011	$\mathbf{0.769} \pm 0.037$
100	7.082 ± 0.407	$\mathbf{6.791} \pm 0.192$	1.551 ± 0.081	$\mathbf{1.384} \pm 0.044$

4.2 Dynamics-Parameter Suboptimality Alters Representation Formation

Both regimes learn structured latent representations, but their latent geometries progressively diverge during training. We quantify this divergence using linear centered kernel alignment (CKA), which measures representation similarity while being invariant to isotropic scaling and orthogonal transformations. CKA between paired Joint and Catch-up representations decreases progressively and remains substantially below one at convergence (Figure 2a), indicating a persistent separation of their representation trajectories.

For visualization, we project a fixed evaluation set into a common PCA space shared across methods and checkpoints, using fixed axes and final-scale alignment. Figure 2b shows a progressive separation of the latent geometries learned by Joint and Catch-up. Together with the update diagnostic above, the PCA and CKA results show that these early update differences are accompanied by progressively diverging representation trajectories.

We next examine whether the divergent representation trajectories correspond to differences in multi-step rollout quality. When the final Joint and Catch-up models are rolled out autoregressively, Catch-up yields lower image-space MSE at every evaluated horizon, with the largest margins at short and intermediate horizons (Figure 3). However, this end-to-end comparison conflates the dynamical quality of the learned representation with final transition fitting quality, precluding attribution of the rollout gap to the representation alone.

To control for unequal transition optimization, we freeze each learned representation–decoder pair and refit a fresh transition model using identical initialization and the same refitting procedure within each Joint–Catch-up pair. Table 1 shows that Catch-up achieves lower latent NMSE at every evaluated horizon, with reductions of 8.7%–27.3% relative to Joint, indicating that its learned representation supports more accurate scale-normalized multi-step rollout; image-space MSE is likewise reduced by 4.1%–16.6%, extending this advantage to observation space. Together, these results

Algorithm 1 Asymmetric Coupling Warm-up (ACW)

Require: Native training procedure; representation parameters ϕ; $k_{\text{start}}, k_{\text{full}}$

for optimization step k **do**
  $\gamma_k \leftarrow \Gamma(k; k_{\text{start}}, k_{\text{full}})$
  Compute $\mathbf{g}_{\phi}^{\text{Native}}$ and $\mathbf{g}_{\phi}^{\text{dyn}}$
  $\mathbf{g}_{\phi}^{\text{ACW}} \leftarrow \mathbf{g}_{\phi}^{\text{Native}} - (1-\gamma_k)\mathbf{g}_{\phi}^{\text{dyn}}$
  Update ϕ using $\mathbf{g}_{\phi}^{\text{ACW}}$, and all other parameters using their Native gradients
end for

show that reducing transition suboptimality during training changes both the trajectory of representation formation and the dynamical quality of the learned representation.

5 Cross-Paradigm Evidence for Dynamics–Representation Coupling Effects

We next examine whether the identified mechanism has observable consequences across different learning paradigms. The substantially larger scale of these cross-paradigm experiments makes Catch-up impractical, since it would require repeated dynamics-only optimization to stabilization between joint updates. We therefore use ACW as a lightweight intervention that directly attenuates the corresponding dynamics-to-representation feedback.

5.1 Asymmetric Coupling Warm-up

The controlled study shows that the effect of dynamics-parameter suboptimality on representation updates is more pronounced during early representation formation and diminishes as dynamics fitting improves. Motivated by this observation, Asymmetric Coupling Warm-up (ACW) attenuates early dynamics-to-representation coupling and gradually restores it to the Native level.

Formally, let $\mathbf{g}_{\phi}^{\text{Native}}$ denote the full Native representation gradient and $\mathbf{g}_{\phi}^{\text{dyn}}$ its dynamics-to-representation component. At optimization step k,

$$\mathbf{g}_{\phi}^{\text{ACW}} = \mathbf{g}_{\phi}^{\text{Native}} - (1-\gamma_k)\mathbf{g}_{\phi}^{\text{dyn}},$$

where

$$\gamma_k = \Gamma(k) = \begin{cases} 0, & k < k_{\text{start}}, \\ \dfrac{k - k_{\text{start}}}{k_{\text{full}} - k_{\text{start}}}, & k_{\text{start}} \le k < k_{\text{full}}, \\ 1, & k \ge k_{\text{full}}. \end{cases}$$

Only the dynamics-to-representation component is attenuated; the dynamics parameters continue to receive their Native gradients.

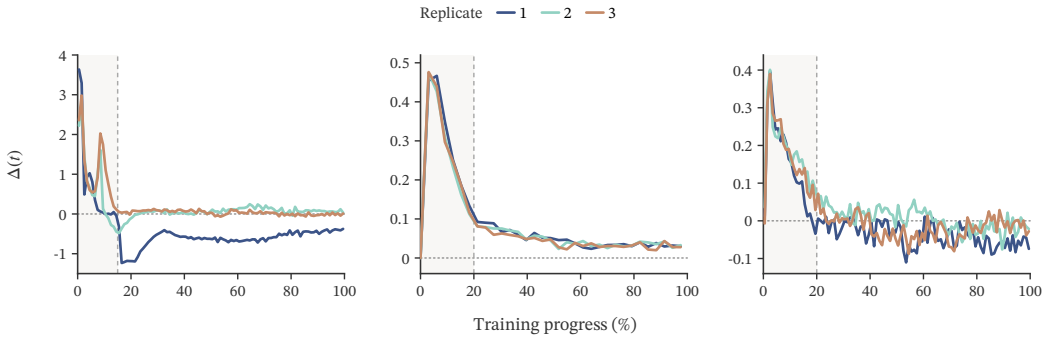
Since isolating the suboptimality-induced component requires a fitted-dynamics reference, ACW instead attenuates the full dynamics-to-representation component early in training, reducing the influence of suboptimality when it is more pronounced. We set k_{start} where the early gradient discrepancy begins to decline and k_{full} where further dynamics-only fitting yields little additional improvement; ACW otherwise retains Native per-step complexity.

5.2 Experimental Setup

We consider Koopman-AE (Gupta et al., 2025), PLDM (Sobal et al., 2025), and STORM (Zhang et al., 2023; 2026). Koopman-AE is studied on two-dimensional flow over a cylinder in a limit-cycle regime, learning a low-dimensional representation together with an explicit linear Koopman operator for multi-step prediction and autonomous rollout. PLDM is evaluated on the Two-Rooms navigation task, where a JEPA-style deterministic latent predictor supports closed-loop goal-reaching planning from top-down image observations. STORM is evaluated on the MsPacman task from the Atari 100K benchmark, where a stochastic Transformer world model is learned from pixel observations

Table 2: Cross-paradigm task-level performance. Mean ± s.d. over three independent seeds. Arrows indicate the favorable direction. Relative gain is computed from the condition means, with positive values denoting improvement. Metrics are comparable only within each paradigm.

Model	Metric	Native	ACW	Relative Gain
Koopman-AE	20-step rollout NMSE ($\times 10^{-4}$) ↓	1.307 ± 0.584	**0.816** ± 0.105	**37.6%**
	1000-step rollout NMSE ($\times 10^{-2}$) ↓	1.332 ± 0.653	**1.161** ± 0.410	**12.8%**
PLDM	Success rate ↑	0.470 ± 0.070	**0.560** ± 0.044	**19.1%**
	Planning MSE ↓	178.87 ± 55.44	**143.72** ± 48.25	**19.6%**
STORM	Greedy evaluation return ↑	1205.3 ± 156.1	**1471.5** ± 314.0	**22.1%**
	Late online return ↑	1176.0 ± 151.1	**1421.2** ± 282.0	**20.8%**



(a) Koopman-AE (b) PLDM (c) STORM

Figure 4: Optimization trajectories across the three paradigms. Each color denotes an independent training replicate initialized with a distinct random seed. For visualization, the Koopman-AE and STORM trajectories are aggregated over 1% training-progress intervals, whereas PLDM is shown at its training-log points.

collected through online interaction and used for model-based control. Together, these settings differ substantially in both dynamics parameterization and downstream use, covering linear latent dynamics, deterministic self-prediction, and stochastic world modeling across prediction, planning, and control. For each paradigm, we preserve its native architecture, objective, task, and evaluation protocol, and compare **Native** training against ACW. Full details are provided in Appendix C.

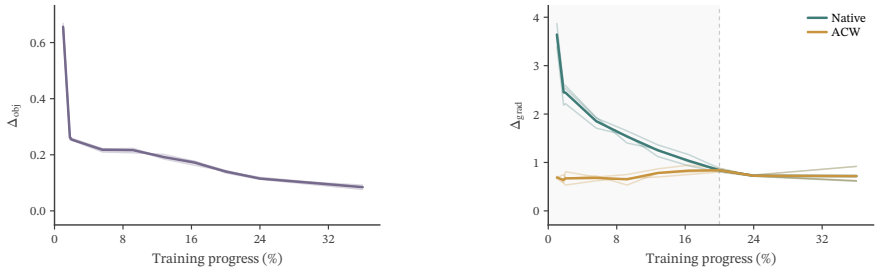
5.3 Cross-Paradigm Outcomes and Optimization Trajectories

Task-level outcomes. Table 2 compares Native and ACW under each paradigm's original task and evaluation protocol. We report two complementary task-level metrics for each model: short- and long-horizon rollout accuracy for Koopman-AE, planning success rate and continuous planning error for PLDM, and fixed-protocol greedy return and late-stage online return for STORM. Across the three training seeds, the overall performance favors ACW on both reported metrics in each paradigm. Taken together, these evaluations span autonomous state prediction, closed-loop planning, and control performance, providing evidence that the consequences of early dynamics-to-representation coupling extend beyond the controlled study in Section 4.

Optimization trajectories. To characterize how Native and ACW differ during training, we track the ACW-to-Native log-ratio

$$\Delta(t) = \log_{10} \frac{L_{\mathrm{ACW}}(t)}{L_{\mathrm{Native}}(t)},$$

where L denotes the Koopman evolution loss, predictor similarity loss, and posterior–prior KL for Koopman-AE, PLDM, and STORM, respectively. Positive values indicate a larger dynamics-fitting quantity under ACW. As shown in Figure 4, the log-ratio is predominantly positive early in training across all three paradigms and is substantially closer to zero later in training. The larger early discrepancy is an intended consequence of ACW: while the dynamics model is still adapting, the



(a) Objective Gap to Reference

(b) Gradient Distance to Reference

Figure 5: Reference-based diagnostic in PLDM. (a) Objective gap between the current predictor and a better-fitted predictor reference under a fixed representation. (b) Representation-gradient distance to the corresponding reference gradient for Native and ACW. Thin, translucent curves denote individual seeds, and thick curves denote the mean across three seeds. The vertical dashed line marks the restoration of full coupling ($\gamma = 1$).

dynamics-to-representation coupling is attenuated, whereas the remaining Native gradient pathways are left unchanged. ACW thus delays the stage at which dynamics-related gradients can fully shape the representation, rather than seeking uniformly lower dynamics-fitting values throughout training. We next examine whether early coupling attenuation reduces the representation-level effect of dynamics-parameter suboptimality.

5.4 Attenuating Suboptimality-Induced Representation-Gradient Effects

To examine whether ACW attenuates suboptimality-induced representation-gradient effects, we conduct a controlled diagnostic in PLDM, whose predictor can be further optimized while holding the representation fixed. At selected checkpoints along a Native training trajectory, we keep $(\boldsymbol{\phi}, \boldsymbol{\psi})$ fixed and evaluate the representation gradient under both Native and ACW. We then freeze $\boldsymbol{\phi}$ and further optimize only $\boldsymbol{\psi}$ to obtain a better-fitted predictor reference $\boldsymbol{\psi}^{\text{ref}}$. Because $\boldsymbol{\psi}^{\text{ref}}$ is obtained by finite predictor-only optimization, we do not treat it as an exact optimum. Using the PLDM predictor-fitting objective for E, we define

$$\Delta_{\text{obj}} = E(\boldsymbol{\phi}, \boldsymbol{\psi}) - E(\boldsymbol{\phi}, \boldsymbol{\psi}_{\text{ref}}), \qquad \Delta^{m}_{\text{grad}} = \left\| \mathbf{g}^{m}_{\boldsymbol{\phi}} - \mathbf{g}^{\text{ref}}_{\boldsymbol{\phi}} \right\|_2, \quad m \in \{\text{Native}, \text{ACW}\}.$$

Here, $\mathbf{g}^{\text{ref}}_{\boldsymbol{\phi}}$ denotes the representation gradient evaluated with $\boldsymbol{\psi}_{\text{ref}}$, and $\mathbf{g}^{m}_{\boldsymbol{\phi}}$ denotes the corresponding gradient under Native or ACW.

Figure 5a confirms that further predictor-only optimization substantially reduces the fitting objective at early checkpoints. Figure 5b further shows that, at these same checkpoints, the representation gradient under ACW is closer to the reference gradient than under Native. Thus, ACW attenuates the representation-gradient effect of predictor suboptimality without requiring additional predictor-only updates. Taken together, these results extend the evidence for the effects of dynamics-parameter suboptimality beyond the controlled study to substantially different representation–dynamics learning paradigms.

6 Conclusion

This work characterizes the representation-gradient discrepancy arising from dynamics-parameter suboptimality and demonstrates its consequences for representation formation and learned dynamical quality. Cross-paradigm evidence further indicates that the identified mechanism arises across substantially different representation–dynamics paradigms. ACW provides a lightweight, mechanism-motivated intervention rather than a general solution, leaving dynamics-parameter suboptimality as an open problem for future investigation. More broadly, under joint optimization, dynamics-induced representation learning reflects not only the dynamics inductive bias, but also the additional influence introduced by dynamics-parameter suboptimality.

AI USE STATEMENT

Generative AI tools were used to assist with code implementation and debugging, literature search and organization, language editing, and limited discussion of mathematical derivations. The research questions, core ideas, theoretical results, experimental design, and conclusions were independently developed by the authors. All AI-assisted mathematical content was independently checked and re-derived by the authors before inclusion. The authors take full responsibility for the final manuscript.

REPRODUCIBILITY STATEMENT

We provide the assumptions, derivations, and complete proofs of the theoretical results in the appendix. The appendix also documents the experimental protocols, model and optimization settings, evaluation procedures, and diagnostic definitions used throughout the paper. The supplementary material includes an anonymous reproduction package covering all four experimental settings, with the source code, frozen configurations, data-preparation procedures, evaluation and aggregation scripts, and executable smoke tests required to reproduce the reported results. For datasets that are not redistributed directly, the supplementary material provides the corresponding acquisition and preprocessing procedures together with integrity checks for the processed data.

REFERENCES

Justin Bayer, Maximilian Soelch, Atanas Mirchev, Baris Kayalibay, and Patrick van der Smagt. Mind the gap when conditioning amortised inference in sequential latent-variable models. In *International Conference on Learning Representations*, 2021.

Xiaoyuan Cheng, Wenxuan Yuan, Yiming Yang, Yuanzhao Zhang, Sibo Cheng, Yi He, and Zhuo Sun. Information shapes Koopman representation. In *International Conference on Learning Representations*, 2026.

John M. Danskin. *The Theory of Max-Min and Its Application to Weapons Allocation Problems*, volume 5 of *Econometrics and Operations Research*. Springer-Verlag, Berlin, 1967. doi: 10.1007/978-3-642-46092-0.

Katharina Friedl, Noémie Jaquier, Jens Lundell, Tamim Asfour, and Danica Kragic. A Riemannian framework for learning reduced-order Lagrangian dynamics. In *International Conference on Learning Representations*, 2025.

Priyam Gupta, Peter Schmid, Denis Sipp, Taraneh Sayadi, and Georgios Rigas. Mori–Zwanzig latent space Koopman closure for nonlinear autoencoder. *Proceedings of the Royal Society A: Mathematical, Physical and Engineering Sciences*, 481(2313):20240259, 2025. doi: 10.1098/rspa.2024.0259.

David Ha and Jürgen Schmidhuber. Recurrent world models facilitate policy evolution. In *Advances in Neural Information Processing Systems*, volume 31, 2018.

Danijar Hafner, Timothy Lillicrap, Ian Fischer, Ruben Villegas, David Ha, Honglak Lee, and James Davidson. Learning latent dynamics for planning from pixels. In *Proceedings of the 36th International Conference on Machine Learning*, volume 97 of *Proceedings of Machine Learning Research*, pp. 2555–2565. PMLR, 2019.

Danijar Hafner, Timothy P. Lillicrap, Mohammad Norouzi, and Jimmy Ba. Mastering Atari with discrete world models. In *International Conference on Learning Representations*, 2021.

Danijar Hafner, Jurgis Pasukonis, Jimmy Ba, and Timothy Lillicrap. Mastering diverse control tasks through world models. *Nature*, 640:647–653, 2025. doi: 10.1038/s41586-025-08744-2.

Maximilian Karl, Maximilian Soelch, Justin Bayer, and Patrick van der Smagt. Deep variational Bayes filters: Unsupervised learning of state space models from raw data. In *International Conference on Learning Representations*, 2017.

Khimya Khetarpal, Zhaohan Daniel Guo, Bernardo Avila Pires, Yunhao Tang, Clare Lyle, Mark Rowland, Nicolas Heess, Diana L Borsa, Arthur Guez, and Will Dabney. A unifying framework for action-conditional self-predictive reinforcement learning. In *Proceedings of The 28th International Conference on Artificial Intelligence and Statistics*, volume 258 of *Proceedings of Machine Learning Research*, pp. 181–189. PMLR, 2025.

Alexej Klushyn, Richard Kurle, Maximilian Soelch, Botond Cseke, and Patrick van der Smagt. Latent matters: Learning deep state-space models. In *Advances in Neural Information Processing Systems*, volume 34, pp. 10234–10245, 2021.

Bethany Lusch, J. Nathan Kutz, and Steven L. Brunton. Deep learning for universal linear embeddings of nonlinear dynamics. *Nature Communications*, 9:4950, 2018. doi: 10.1038/s41467-018-07210-0.

Max Schwarzer, Ankesh Anand, Rishab Goel, R Devon Hjelm, Aaron Courville, and Philip Bachman. Data-efficient reinforcement learning with self-predictive representations. In *International Conference on Learning Representations*, 2021.

Younggyo Seo, Kimin Lee, Stephen L. James, and Pieter Abbeel. Reinforcement learning with action-free pre-training from videos. In *Proceedings of the 39th International Conference on Machine Learning*, volume 162 of *Proceedings of Machine Learning Research*, pp. 19561–19579. PMLR, 2022.

Uladzislau Sobal, Wancong Zhang, Kyunghyun Cho, Randall Balestriero, Tim G. J. Rudner, and Yann LeCun. Learning from reward-free offline data: A case for planning with latent dynamics models. In *Advances in Neural Information Processing Systems*, volume 38, 2025. doi: 10.52202/085713-1465.

Naoya Takeishi, Yoshinobu Kawahara, and Takehisa Yairi. Learning Koopman invariant subspaces for dynamic mode decomposition. In *Advances in Neural Information Processing Systems*, volume 30, 2017.

Yunhao Tang, Zhaohan Daniel Guo, Pierre Harvey Richemond, Bernardo Avila Pires, Yash Chandak, Remi Munos, Mark Rowland, Mohammad Gheshlaghi Azar, Charline Le Lan, Clare Lyle, András György, Shantanu Thakoor, Will Dabney, Bilal Piot, Daniele Calandriello, and Michal Valko. Understanding self-predictive learning for reinforcement learning. In *Proceedings of the 40th International Conference on Machine Learning*, volume 202 of *Proceedings of Machine Learning Research*, pp. 33632–33656. PMLR, 2023.

Giacomo Turri, Luigi Bonati, Kai Zhu, Massimiliano Pontil, and Pietro Novelli. Self-supervised evolution operator learning for high-dimensional dynamical systems. In *International Conference on Learning Representations*, 2026.

Yucen Wang, Fengming Zhang, De-Chuan Zhan, Li Zhao, Kaixin Wang, and Jiang Bian. Co-evolving latent action world models. In *Proceedings of the 43rd International Conference on Machine Learning*, volume 306 of *Proceedings of Machine Learning Research*. PMLR, 2026.

Weipu Zhang, Gang Wang, Jian Sun, Yetian Yuan, and Gao Huang. STORM: Efficient stochastic transformer based world models for reinforcement learning. In *Advances in Neural Information Processing Systems*, volume 36, 2023. doi: 10.52202/075280-1182.

Weipu Zhang, Adam Jelley, Trevor McInroe, Amos Storkey, and Gang Wang. Object-centric world models from few-shot annotations for sample-efficient reinforcement learning. In *International Conference on Learning Representations*, 2026.

Gaoyue Zhou, Hengkai Pan, Yann LeCun, and Lerrel Pinto. DINO-WM: World models on pre-trained visual features enable zero-shot planning. In *Proceedings of the 42nd International Conference on Machine Learning*, volume 267 of *Proceedings of Machine Learning Research*, pp. 79115–79135. PMLR, 2025.

A Theoretical Details and Derivations

A.1 Proof of Proposition 1

Since the dynamics-fitting minimum is attained, define

$$\Psi^\star(\boldsymbol{\phi}) := \arg\min_{\boldsymbol{\psi}\in\Psi} E(\boldsymbol{\phi},\boldsymbol{\psi}),$$

which is nonempty. By the directional form of Danskin's theorem, for every $\boldsymbol{d} \in \mathbb{R}^{d_\phi}$,

$$A'_{\mathcal{P}}(\boldsymbol{\phi};\boldsymbol{d}) = \min_{\boldsymbol{\psi}^\star\in\Psi^\star(\boldsymbol{\phi})} \langle \nabla_{\boldsymbol{\phi}} E(\boldsymbol{\phi},\boldsymbol{\psi}^\star), \boldsymbol{d}\rangle .$$

Since $A_{\mathcal{P}}$ is differentiable at $\boldsymbol{\phi}$,

$$A'_{\mathcal{P}}(\boldsymbol{\phi};\boldsymbol{d}) = \langle \nabla_{\boldsymbol{\phi}} A_{\mathcal{P}}(\boldsymbol{\phi}), \boldsymbol{d}\rangle .$$

Therefore,

$$\langle \nabla_{\boldsymbol{\phi}} A_{\mathcal{P}}(\boldsymbol{\phi}), \boldsymbol{d}\rangle = \min_{\boldsymbol{\psi}^\star\in\Psi^\star(\boldsymbol{\phi})} \langle \nabla_{\boldsymbol{\phi}} E(\boldsymbol{\phi},\boldsymbol{\psi}^\star), \boldsymbol{d}\rangle .$$

Hence, for any fitted optimum

$$\boldsymbol{\psi}^\star_{\mathrm{fit}}(\boldsymbol{\phi}) \in \Psi^\star(\boldsymbol{\phi}),$$

we have

$$\langle \nabla_{\boldsymbol{\phi}} A_{\mathcal{P}}(\boldsymbol{\phi}), \boldsymbol{d}\rangle \le \langle \nabla_{\boldsymbol{\phi}} E(\boldsymbol{\phi}, \boldsymbol{\psi}^\star_{\mathrm{fit}}(\boldsymbol{\phi})), \boldsymbol{d}\rangle \qquad \forall\, \boldsymbol{d} \in \mathbb{R}^{d_\phi}.$$

Let

$$\boldsymbol{v} := \nabla_{\boldsymbol{\phi}} E(\boldsymbol{\phi}, \boldsymbol{\psi}^\star_{\mathrm{fit}}(\boldsymbol{\phi})) - \nabla_{\boldsymbol{\phi}} A_{\mathcal{P}}(\boldsymbol{\phi}).$$

Then

$$\langle \boldsymbol{v}, \boldsymbol{d}\rangle \ge 0 \qquad \forall\, \boldsymbol{d} \in \mathbb{R}^{d_\phi}.$$

Choosing $\boldsymbol{d} = -\boldsymbol{v}$ gives

$$-\|\boldsymbol{v}\|_2^2 \ge 0,$$

and therefore $\boldsymbol{v} = \mathbf{0}$. Thus, for any fitted optimum,

$$\nabla_{\boldsymbol{\phi}} A_{\mathcal{P}}(\boldsymbol{\phi}) = \nabla_{\boldsymbol{\phi}} E(\boldsymbol{\phi}, \widetilde{\boldsymbol{\psi}})\Big|_{\widetilde{\boldsymbol{\psi}}=\boldsymbol{\psi}^\star_{\mathrm{fit}}(\boldsymbol{\phi})},$$

where $\widetilde{\boldsymbol{\psi}}$ is held fixed when differentiating with respect to $\boldsymbol{\phi}$. This proves Proposition 1. □

As an immediate consequence, using

$$O_{\mathcal{P}}(\boldsymbol{\phi},\boldsymbol{\psi}) = E(\boldsymbol{\phi},\boldsymbol{\psi}) - A_{\mathcal{P}}(\boldsymbol{\phi}),$$

we further obtain

$$\begin{aligned}\nabla_{\boldsymbol{\phi}} O_{\mathcal{P}}(\boldsymbol{\phi},\boldsymbol{\psi}) &= \nabla_{\boldsymbol{\phi}} E(\boldsymbol{\phi},\boldsymbol{\psi}) - \nabla_{\boldsymbol{\phi}} A_{\mathcal{P}}(\boldsymbol{\phi}) \\ &= \nabla_{\boldsymbol{\phi}} E(\boldsymbol{\phi},\boldsymbol{\psi}) - \nabla_{\boldsymbol{\phi}} E(\boldsymbol{\phi}, \widetilde{\boldsymbol{\psi}})\Big|_{\widetilde{\boldsymbol{\psi}}=\boldsymbol{\psi}^\star_{\mathrm{fit}}(\boldsymbol{\phi})}.\end{aligned}$$

Hence, when $\mathbf{g}^{\mathrm{dyn}}_{\boldsymbol{\phi}} = \nabla_{\boldsymbol{\phi}} E$, the current-versus-fitted representation-gradient discrepancy satisfies

$$\Delta \mathbf{g}_{\boldsymbol{\phi}} = \nabla_{\boldsymbol{\phi}} O_{\mathcal{P}}(\boldsymbol{\phi},\boldsymbol{\psi}),$$

as stated in the main text.

A.2 Proof of Proposition 2

Using the temporal factorizations of the posterior and latent prior,

$$q_{\boldsymbol{\phi}}(\mathbf{z} \mid \mathbf{x}) = \prod_{t=1}^{T} q_{\boldsymbol{\phi}}(z_t \mid z_{<t}, \mathbf{x}), \qquad p_{\boldsymbol{\psi}}(\mathbf{z}) = \prod_{t=1}^{T} p_{\boldsymbol{\psi}}(z_t \mid z_{<t}),$$

the rate can be written as

$$R(\boldsymbol{\phi}, \boldsymbol{\psi}) = \mathbb{E}_{\mathbf{x}\sim p_{\mathrm{data}},\, \mathbf{z}\sim q_{\boldsymbol{\phi}}(\cdot|\mathbf{x})} \left[\log \frac{q_{\boldsymbol{\phi}}(\mathbf{z} \mid \mathbf{x})}{p_{\boldsymbol{\psi}}(\mathbf{z})}\right] = \sum_{t=1}^{T} \mathbb{E}_{r_{\boldsymbol{\phi}}(\mathbf{x},\mathbf{z})} \left[\log \frac{q_{\boldsymbol{\phi}}(z_t \mid z_{<t}, \mathbf{x})}{p_{\boldsymbol{\psi}}(z_t \mid z_{<t})}\right].$$

For each time step, insert the data-aggregated posterior conditional $r_{\boldsymbol{\phi}}(z_t \mid z_{<t})$:

$$\log \frac{q_{\boldsymbol{\phi}}(z_t \mid z_{<t}, \mathbf{x})}{p_{\boldsymbol{\psi}}(z_t \mid z_{<t})} = \log \frac{q_{\boldsymbol{\phi}}(z_t \mid z_{<t}, \mathbf{x})}{r_{\boldsymbol{\phi}}(z_t \mid z_{<t})} + \log \frac{r_{\boldsymbol{\phi}}(z_t \mid z_{<t})}{p_{\boldsymbol{\psi}}(z_t \mid z_{<t})}.$$

By the definition $r_{\boldsymbol{\phi}}(\mathbf{x}, \mathbf{z}) = p_{\mathrm{data}}(\mathbf{x}) q_{\boldsymbol{\phi}}(\mathbf{z} \mid \mathbf{x})$, we have

$$r_{\boldsymbol{\phi}}(z_t \mid z_{<t}, \mathbf{x}) = q_{\boldsymbol{\phi}}(z_t \mid z_{<t}, \mathbf{x}).$$

Therefore, the expectation of the first term is

$$\mathbb{E}_{r_{\boldsymbol{\phi}}} \left[\log \frac{q_{\boldsymbol{\phi}}(z_t \mid z_{<t}, \mathbf{x})}{r_{\boldsymbol{\phi}}(z_t \mid z_{<t})}\right] = \mathbb{E}_{r_{\boldsymbol{\phi}}} \left[\log \frac{r_{\boldsymbol{\phi}}(z_t \mid z_{<t}, \mathbf{x})}{r_{\boldsymbol{\phi}}(z_t \mid z_{<t})}\right] = I_{r_{\boldsymbol{\phi}}}(Z_t; X \mid Z_{<t}).$$

The expectation of the second term is

$$\mathbb{E}_{r_{\boldsymbol{\phi}}} \left[\log \frac{r_{\boldsymbol{\phi}}(z_t \mid z_{<t})}{p_{\boldsymbol{\psi}}(z_t \mid z_{<t})}\right] = \mathbb{E}_{z_{<t}\sim r_{\boldsymbol{\phi}}} D_{\mathrm{KL}} \left(r_{\boldsymbol{\phi}}(z_t \mid z_{<t}) \parallel p_{\boldsymbol{\psi}}(z_t \mid z_{<t})\right).$$

Summing over $t = 1, \ldots, T$ gives

$$\begin{aligned} R(\boldsymbol{\phi}, \boldsymbol{\psi}) &= \sum_{t=1}^{T} I_{r_{\boldsymbol{\phi}}}(Z_t; X \mid Z_{<t}) \\ &\quad + \sum_{t=1}^{T} \mathbb{E}_{z_{<t}\sim r_{\boldsymbol{\phi}}} D_{\mathrm{KL}} \left(r_{\boldsymbol{\phi}}(z_t \mid z_{<t}) \parallel p_{\boldsymbol{\psi}}(z_t \mid z_{<t})\right) \\ &= C(\boldsymbol{\phi}) + E(\boldsymbol{\phi}, \boldsymbol{\psi}). \end{aligned}$$

For $t = 1$, $Z_{<1}$ is empty, so the corresponding terms reduce to

$$I_{r_{\boldsymbol{\phi}}}(Z_1; X) \qquad \text{and} \qquad D_{\mathrm{KL}} \left(r_{\boldsymbol{\phi}}(z_1) \parallel p_{\boldsymbol{\psi}}(z_1)\right),$$

respectively, with the latter corresponding to the aggregated initial posterior and the initial prior. This proves Proposition 2. □

B Controlled Study

B.1 Controlled Study Setup

Data. We use a simulated double-pendulum system with state $s_t = (\theta_1, \theta_2, \omega_1, \omega_2)$, unit masses and link lengths, gravity $g = 9.81$, and linear damping torques $-0.01\omega_i$. Trajectories are integrated using fixed-step RK4 with step size 0.005, after a 0.5-unit burn-in, and observations are recorded every 0.04 time units. Initial conditions are sampled independently as

$$\theta_1 \sim \mathcal{U}(0.72\pi, 1.28\pi), \qquad \theta_2 \sim \mathcal{U}(-0.55\pi, 0.55\pi), \qquad \omega_1, \omega_2 \sim \mathcal{U}(-0.6, 0.6).$$

Each trajectory contains 128 observations. We use 512/64/64 trajectories for training, validation, and testing, respectively. States are deterministically rendered as 64×64 grayscale images in $[0, 1]$; no observation noise, augmentation, or dataset-level normalization is used. Training operates on complete trajectories rather than temporal crops.

Table 3: Architecture of the sequential VAE used in the controlled study. All hidden nonlinearities are SiLU.

Component	Layer	#	Dimensions	Activation
Posterior	Conv2d 3×3, stride 2	3	$1 \to 8 \to 16 \to 32$	SiLU
	Flatten + Linear	1	$32 \times 8 \times 8 \to 96$	SiLU
	GRU	1	$96 \to 64$, context $= 8$	–
	Mean / log-std heads	2	$64 \to 8$	–
Transition	GRU	1	$8 \to 64$, history ≤ 8	–
	Fully connected	1	$64 \to 64$	SiLU
	Mean head	1	$64 \to 8$	–
Decoder	Fully connected	1	$8 \to 192$	SiLU
	Fully connected	1	$192 \to 384$	SiLU
	Fully connected	1	$384 \to 4096$	Sigmoid

Model architecture. The posterior uses the most recent eight observations, while the transition uses up to eight preceding latent states. The latent dimension is $d_z = 8$. Table 3 summarizes the architecture.

The posterior is diagonal Gaussian. The transition predicts its conditional mean with fixed standard deviation $\sigma_p = 0.05$, and the decoder output parameterizes a factorized Gaussian observation model with fixed standard deviation $\sigma_x = 0.08$. At the beginning of a trajectory, the posterior context is padded with the first-frame feature, while the transition uses all available latent history until eight preceding states are available.

Objective and optimization. Training minimizes

$$\mathcal{L}_\beta = D + \beta \left(R_{\text{init}} + R_{\text{tr}} \right), \qquad \beta = 0.05,$$

where D is the Gaussian observation negative log-likelihood, summed over pixels within each frame and averaged over batch and time,

$$R_{\text{init}} = \frac{1}{T} D_{\text{KL}}(q_\phi(z_1 \mid x) \,\|\, \mathcal{N}(0, I)),$$

and

$$R_{\text{tr}} = \frac{1}{T} \sum_{t=2}^{T} \mathbb{E}[D_{\text{KL}}(q_\phi(z_t \mid x) \,\|\, p_\psi(z_t \mid z_{<t}))].$$

The transition history is formed from reparameterized posterior samples, and R_{tr} is differentiated through the posterior. No KL annealing, auxiliary objective, or additional latent regularization is used.

Training uses batches of 32 complete trajectories for 10,000 outer updates. Posterior and decoder parameters share one Adam optimizer, while transition parameters use a second Adam optimizer. Both use learning rate 5×10^{-3}, Adam coefficients $(0.9, 0.999)$, $\epsilon = 10^{-8}$, and zero weight decay. Gradients are clipped to global norm 5. We use three matched training seeds: 1101, 2202, and 3303. Within each seed, paired Joint and Catch-up runs use the same model seed, outer minibatches, and posterior reparameterization noise at every outer step.

B.2 Joint and Catch-up Training

Joint training directly optimizes the full β-weighted negative ELBO in Appendix B.1, jointly updating the posterior, decoder, and transition model at each outer step. Catch-up performs the same outer update and then carries out additional transition-only optimization while holding the posterior and decoder fixed. The same persistent transition Adam optimizer is used for the outer and inner updates; its learning rate is temporarily reduced from 5×10^{-3} to 10^{-3} during the inner phase.

After each outer update, the 512 training trajectories are randomly partitioned into a 256-trajectory inner pool and a disjoint 256-trajectory monitor set. Posterior statistics and pre-sampled latent histories are cached and detached before the inner optimization. The inner objective is the transition-parameter-dependent part of βR_{tr}, excluding reconstruction and the initial-prior term. Transition-only updates use minibatches of 128 cached trajectories, with gradients clipped to norm 5.

The number of additional transition updates is determined adaptively using the monitor set, which is held fixed throughout the corresponding inner phase. Every eight inner updates, we evaluate

$$R_j = R_{\mathrm{init}} + R_{\mathrm{tr},j}$$

on the cached monitor set and compute

$$\delta_j = \frac{|R_j - R_{j-1}|}{|R_{j-1}| + 10^{-12}}.$$

After at least 16 inner updates, Catch-up terminates when $\delta_j < 0.01$ for three consecutive monitoring points, with a maximum budget of 256 inner updates. Thus Joint and Catch-up use the same number of outer posterior and decoder updates; Catch-up differs by allowing the transition model additional adaptation before the next outer update.

B.3 Reference-Based Diagnostics

We perform the reference-based diagnostic at outer steps $100, 200, 400, 2000$, and 10000 for all three seeds and both Joint and Catch-up. For each seed–checkpoint pair, Joint and Catch-up use the same randomly generated audit pools.

Better-fitted transition reference. For each checkpoint, we sample a fixed audit pool of $65{,}536$ trajectory–time windows from the training set and cache the corresponding posterior quantities. Transition histories are formed from fixed reparameterized posterior samples, while posterior means serve as fitting targets. Starting from the checkpointed transition parameters and the saved transition-optimizer state, we perform 2048 additional transition-only updates with minibatches of 4096 cached windows. The learning rate is 10^{-3} for the first 1024 updates and 2×10^{-4} thereafter; transition gradients are clipped to norm 5.

The transition-fitting objective is evaluated on the full fixed audit pool every 64 updates, including the initial state at $K = 0$, and the lowest-objective monitored state is retained as $\boldsymbol{\psi}_{\mathrm{ref}}$. Thus, $\boldsymbol{\psi}_{\mathrm{ref}}$ is a better-fitted finite-budget reference rather than an exact fitted optimum.

Using the same audit pool, we report

$$\Delta E_{\mathrm{fit}} = E_{\mathrm{fit}}(\boldsymbol{\phi}, \boldsymbol{\psi}_{\mathrm{cur}}) - E_{\mathrm{fit}}(\boldsymbol{\phi}, \boldsymbol{\psi}_{\mathrm{ref}}),$$

where $\boldsymbol{\psi}_{\mathrm{cur}}$ is the transition stored in the original checkpoint. Here E_{fit} denotes the transition-dependent part of the rate, evaluated before multiplication by β.

Adam-level representation-update discrepancy. We additionally sample a fixed pool of 64 complete training trajectories and fixed posterior noise. Under $\boldsymbol{\psi}_{\mathrm{cur}}$ and $\boldsymbol{\psi}_{\mathrm{ref}}$, respectively, we compute gradients of the complete training objective $D + \beta(R_{\mathrm{init}} + R_{\mathrm{tr}})$. For each case, we reproduce the global gradient clipping used during training and apply the same checkpointed representation-optimizer state to obtain the corresponding next posterior Adam displacement, $\Delta\boldsymbol{\phi}_{\mathrm{cur}}$ and $\Delta\boldsymbol{\phi}_{\mathrm{ref}}$. We report

$$\rho_{\mathrm{Adam}} = \frac{\|\Delta\boldsymbol{\phi}_{\mathrm{cur}} - \Delta\boldsymbol{\phi}_{\mathrm{ref}}\|_2}{\max(\|\Delta\boldsymbol{\phi}_{\mathrm{ref}}\|_2, 10^{-12})}.$$

Thus, ρ_{Adam} measures the discrepancy between the next-step optimizer-level posterior updates rather than between raw gradients. Figure 1 reports ΔE_{fit} and ρ_{Adam} across the three matched seeds.

B.4 Representation-Trajectory Diagnostics

Linear CKA. During training, we save latent probes on a fixed evaluation set containing all 64 validation trajectories and all 128 time points. Each probe uses the posterior mean and therefore

has shape $64 \times 128 \times 8$. For each matched Joint–Catch-up pair, the latent arrays are reshaped to 8192×8, centered independently along each latent dimension, and compared using linear centered CKA:

$$\mathrm{CKA}(X, Y) = \frac{\left\|X^\top Y\right\|_F^2}{\sqrt{\|X^\top X\|_F^2 \, \|Y^\top Y\|_F^2}}.$$

The denominator is clamped below by 10^{-12}. We apply no whitening or additional feature normalization.

We evaluate 57 predetermined probe steps: step 0; every 25 steps from 25 to 500; every 50 steps from 550 to 1000; every 100 steps from 1100 to 2000; and every 500 steps from 2500 to 10000. CKA is computed separately for each matched seed and then averaged across the three seeds. Figure 2a shows the resulting representation-similarity trajectories.

Shared PCA visualization. For the latent-geometry visualization in Figure 2b, we use seed 1101 and posterior-mean latent clouds from Joint and Catch-up at outer steps

$$0,\ 75,\ 125,\ 175,\ 300,\ 1000,\ 10000.$$

Each cloud is first centered independently. We then apply final-scale alignment: all Joint clouds are divided by the RMS radius of the final Joint cloud, and all Catch-up clouds by the RMS radius of the final Catch-up cloud. The resulting 14 aligned clouds are concatenated, and a single three-dimensional PCA basis and common center are fitted to the combined data. The same PCA coordinates are therefore used across methods and checkpoints.

The displayed panels correspond to steps $75, 175, 1000$, and 10000. Points are colored by the first-link angle θ_1, wrapped to $(-\pi, \pi]$. The PCA panels provide a visualization of the evolving latent geometry, while linear CKA is used as the quantitative measure of representation similarity.

B.5 Rollout Evaluation

Original-model rollout. We evaluate the final Joint and Catch-up models on all 64 test trajectories. For each trajectory, the first eight posterior means initialize the latent history. Subsequent latent states are generated autonomously using the transition mean and recursively fed back into the transition model; the posterior receives no further observations during rollout. Predictions are decoded to image space and evaluated at horizons $H \in \{10, 25, 50, 100\}$.

For each horizon, image MSE is computed over all predicted frames up to H, all test trajectories, and all pixels:

$$\mathrm{MSE}_H = \mathrm{mean}_{i,t \le H,p} \left(\hat{x}_{i,t,p} - x_{i,t,p}\right)^2,$$

where the eight initialization frames are excluded from the error. The images lie in $[0, 1]$, and no additional normalization is applied. Figure 3 reports the three individual seeds together with their arithmetic mean.

Matched transition refit. To separate representation quality from differences in the transition model obtained during the original training run, we freeze each learned posterior–decoder pair and fit a fresh transition model of the same architecture. For each representation seed, we use two paired refit restarts. Within each restart, Joint and Catch-up use identical fresh transition initialization and the same minibatch and posterior-noise schedules, while retaining their respective learned posterior–decoder pairs.

Refitting optimizes only the time-weighted transition-dependent posterior KL using cached posterior statistics from the training split. We use Adam with batch size 128, initial learning rate 0.012, and gradient clipping at norm 5. A fixed-noise validation objective is evaluated every 100 updates and used for checkpoint selection and learning-rate scheduling. The learning rate is reduced by a factor of 0.3 after 10 validation checks without sufficient improvement, down to 10^{-5}. Refitting runs for at least 8,000 and at most 60,000 updates, and terminates after 30 validation checks without a relative improvement of 10^{-4}. The best validation checkpoint under this criterion is retained; test data are not used for model selection.

Table 4: Architecture of the Koopman-AE used for the cylinder-flow experiment. Hidden MLP layers use SELU activations; the encoder and decoder output layers are linear.

Component	Type	Dimensions
Encoder	MLP	$3872 \to 512 \to 256 \to 128 \to 64 \to 2$
Koopman	Linear operator	$2 \to 2$
Decoder	MLP	$2 \to 64 \to 128 \to 256 \to 512 \to 3872$

The refitted models are evaluated using the same autonomous rollout protocol and horizons as above. In addition to image MSE, we report latent normalized MSE. Let $z_{i,t}$ denote the posterior-mean latent target, $\hat{z}_{i,t}$ the corresponding autonomous rollout prediction, and c the mean target latent over all test trajectories and the full 100-step rollout target span for the corresponding representation. We compute

$$\mathrm{NMSE}_H = \frac{\sum_{i,t\le H} \|\hat{z}_{i,t} - z_{i,t}\|_2^2}{\max\left(\sum_{i,t\le H} \|z_{i,t} - c\|_2^2, 10^{-12}\right)}.$$

For each representation seed, metrics are first averaged over the two paired refit restarts. Table 1 then reports the mean $\pm$ sample s.d. across the three representation seeds.

C Cross-Paradigm Evaluation: Experimental Details

C.1 Koopman-AE

Data and architecture. We use the cylinder-flow limit-cycle task (Case I) of Gupta et al. (2025). The released data consist of two-component velocity fields on a 63×63 spatial grid. For computational efficiency, we spatially downsample each velocity component to 44×44 and concatenate the two components into a 3872-dimensional state. We use the 2000-snapshot limit-cycle sequence and form 1000 training and 1000 test snapshots by alternating samples, such that consecutive samples within either subsequence are separated by $0.2D/U_\infty$. No input normalization is applied.

Table 4 summarizes the model architecture. The encoder maps the velocity field to a two-dimensional latent state, which is evolved by a single shared Koopman operator $K \in \mathbb{R}^{2\times 2}$. The operator is a full dense linear matrix without bias or an explicit spectral constraint.

Objective and optimization. For a batch of initial states, let $z_t = E_\phi(x_t)$. The Koopman evolution loss evaluates a 20-step latent rollout,

$$\mathcal{L}_K = \frac{1}{B\eta r} \sum_{i=1}^{B} \sum_{j=1}^{\eta} \left\| E_\phi(x_{t_i}) K^j - E_\phi(x_{t_i+j}) \right\|_2^2, \qquad \eta = 20, \quad r = 2.$$

The reconstruction loss is evaluated over the 20 states in each training sample,

$$\mathcal{L}_{\mathrm{AE}} = \frac{1}{B\eta d} \sum_{i=1}^{B} \sum_{j=0}^{\eta-1} \|D(E_\phi(x_{t_i+j})) - x_{t_i+j}\|_2^2, \qquad d = 3872,$$

and the total training objective is

$$\mathcal{L}_{\mathrm{total}} = 0.1\,\mathcal{L}_K + 100\,\mathcal{L}_{\mathrm{AE}}.$$

Under Native training, the full gradient of $\mathcal{L}_K$ is propagated to both the encoder and K. Under ACW, all gradient paths from $\mathcal{L}_K$ to the encoder, through both the rollout initial latent and the encoded future targets, are multiplied by γ_k, while K continues to receive the full dynamics gradient. The decoder and reconstruction branch are unchanged. For the formal runs,

$$\gamma_k = \begin{cases} 0, & k \le 7440, \\ \dfrac{k - 7440}{74400 - 7440}, & 7440 < k < 74400, \\ 1, & k \ge 74400, \end{cases}$$

Table 5: Extended Koopman-AE evaluation at the final checkpoint. Values are mean ± sample s.d. over three independent seeds. Lower is better for all metrics. The final two rows are the metrics reported in the main-text cross-paradigm comparison.

Metric	Native	ACW
Reconstruction NMSE ($\times 10^{-5}$)	7.822 ± 6.358	$\mathbf{4.351 \pm 2.168}$
One-step state NMSE ($\times 10^{-5}$)	8.295 ± 6.275	$\mathbf{4.268 \pm 2.127}$
One-step latent relative MSE ($\times 10^{-5}$)	1.105 ± 1.329	$\mathbf{0.211 \pm 0.050}$
Mean state MSE, horizons 1–20 ($\times 10^{-6}$)	2.709 ± 1.632	$\mathbf{1.286 \pm 0.279}$
20-step rollout NMSE ($\times 10^{-4}$)	1.307 ± 0.584	$\mathbf{0.816 \pm 0.105}$
1000-step rollout NMSE ($\times 10^{-2}$)	1.332 ± 0.653	$\mathbf{1.161 \pm 0.410}$

where k denotes optimizer step. Relative to the complete 8000-epoch training run, this corresponds to full attenuation for approximately the first 1.5% of training and linear restoration of the coupling through 15%.

Both conditions use Adam with learning rate 5×10^{-5}, weight decay 10^{-5}, batch size 16, and 8000 epochs. The formal runs use random seeds 99, 100, and 101. Evaluation is performed at the final checkpoint, with no early stopping or validation-based checkpoint selection. All scalar results below are reported as mean ± sample standard deviation over these three seeds.

Full predictive evaluation. Table 5 reports an extended evaluation of the same formal Native and ACW runs used in the main text. In addition to the two rollout metrics retained in the main comparison, we report reconstruction accuracy, one-step prediction in state and latent space, and prediction error aggregated over horizons 1–20.

For state-space NMSE metrics, the corresponding state-space MSE is normalized by the element-wise variance of the held-out test sequence. The one-step latent relative MSE instead normalizes latent prediction MSE by the mean squared magnitude of the target latent representation.

For multi-step prediction, a test state is encoded once and subsequently evolved autonomously in latent space,

$$\hat{x}_{t+h} = D\big(E_\phi(x_t)K^h\big),$$

without conditioning on intermediate ground-truth observations. The 20-step evaluation aggregates over all valid starting points in the held-out sequence, while the long rollout repeatedly applies the same latent operator over the full evaluation trajectory.

The additional metrics in Table 5 show that ACW has a lower three-seed mean not only on the two rollout metrics retained in the main text, but also on reconstruction, one-step state prediction, one-step latent prediction, and prediction averaged over the short rollout horizon. Figure 6 reports the full horizon-dependent state NMSE from the same formal evaluations rather than only the endpoint used in the main text.

The optimization trajectory in Figure 4 uses the unweighted Koopman evolution loss $\mathcal{L}_K$. For each seed, Native and ACW losses are averaged separately within non-overlapping 1% training-progress intervals, after which the plotted quantity is

$$\log_{10} \frac{\overline{\mathcal{L}}_K^{\mathrm{ACW}}}{\overline{\mathcal{L}}_K^{\mathrm{Native}}}.$$

The three seeds are shown separately, with no additional smoothing or interpolation.

Koopman-operator structure. As a Koopman-specific structural diagnostic, we additionally report the spectral radius of the learned operator. At the final checkpoint,

$$\rho(K) = 1.0000206 \pm 1.8951 \times 10^{-5} \qquad \text{for Native,}$$

and

$$\rho(K) = 1.0000317 \pm 2.9039 \times 10^{-5} \qquad \text{for ACW,}$$

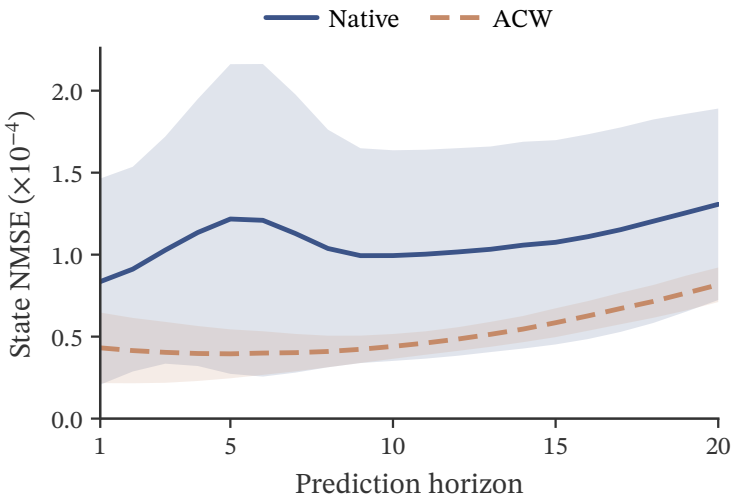


Figure 6: Koopman-AE prediction error over horizons 1–20. Curves show the mean state NMSE over seeds 99, 100, and 101, with shaded regions denoting ± 1 sample s.d. Each rollout is initialized from a ground-truth state once and then evolved autonomously in latent space.

Table 6: Architecture of the PLDM model executed in our Two-Rooms experiments. The final LayerNorm is shared between the encoder and predictor.

Component	**Layer**	**#**	**Dimensions**	**Activation / note**
Encoder	Impala residual stack	3	$2 \to 16 \to 32 \to 32$	ReLU
	Residual blocks / stack	2	Conv 3×3	ReLU
	Flatten + Linear	1	$2592 \to 512$	–
	LayerNorm	1	512	shared
Predictor	GRU	1	action $2 \to$ hidden 512	1 layer
	LayerNorm	1	512	shared with encoder

where the statistics are again computed over seeds 99, 100, and 101. Under both training conditions, the learned operator therefore has a spectral radius extremely close to unity, consistent with the non-decaying dynamics expected in the limit-cycle regime. We use this result as a descriptive Koopman-specific structural diagnostic rather than as a quantity on which ACW is expected to improve over Native training.

C.2 PLDM

Experimental setting. We evaluate PLDM (Sobal et al., 2025) on the Two-Rooms dataset-size condition released as `ds5078`. The archived dataset used in our experiments contains 58 trajectories with 90 stored observation–action rows per trajectory (5,220 rows in total), from which 4,350 length-16 training windows are constructed. Observations are two-channel 65×65 top-down images and actions are two-dimensional continuous displacements. Because the architecture and configuration executed by the released implementation differ in several details from the corresponding prose description in the original paper, we report below the configuration actually executed in our experiments. We use seeds 101, 102, and 103 throughout the PLDM study; the Native–ACW comparison, optimization trajectories, and reference-based mechanism diagnostic all use these same three seeds.

Model architecture. The executed model contains a 512-dimensional visual representation and a single deterministic recurrent predictor. Table 6 summarizes the architecture.

The encoded state $z_0 = h_{\boldsymbol{\phi}}(s_0)$ initializes the GRU hidden state, after which the two-dimensional actions are supplied directly as recurrent inputs. The executed configuration uses one predictor ($K = 1$); there is no predictor ensemble, EMA target encoder, or additional projection head.

Training objective and ACW implementation. For encoded targets $z_{1:H}$ and autoregressive predictions $\hat{z}_{1:H}$, the predictor-fitting term is

$$\mathcal{L}_{\text{sim}} = \text{mean}_{t,b,d} \left(\hat{z}_{t,b,d} - z_{t,b,d}\right)^2 .$$

The complete objective in the executed configuration is

$$\mathcal{L}_{\text{PLDM}} = \mathcal{L}_{\text{sim}} + 2.2\,\mathcal{L}_{\text{var}} + 13\,\mathcal{L}_{\text{cov}} + 0.19\,\mathcal{L}_{\text{time-var}} + 0.9\,\mathcal{L}_{\text{time-sim}},$$

with the temporal-covariance and inverse-dynamics terms disabled. Accordingly, we identify $E_{\text{fit}} = \mathcal{L}_{\text{sim}}$; the remaining active terms provide representation-side gradient pathways that ACW leaves unchanged.

In Native training, $\mathcal{L}_{\text{sim}}$ reaches the encoder through both the predicted branch,

$$\mathcal{L}_{\text{sim}} \to \hat{z}_{1:H} \to z_0 \to h_{\boldsymbol{\phi}},$$

and the encoded target branch,

$$\mathcal{L}_{\text{sim}} \to z_{1:H} \to h_{\boldsymbol{\phi}}.$$

ACW attenuates both encoder-side pathways using

$$\mathcal{R}_\gamma(v) = \text{sg}(v) + \gamma\big(v - \text{sg}(v)\big),$$

which leaves the forward value unchanged while multiplying the backward gradient by γ. Hence, for encoder-exclusive parameters,

$$\mathbf{g}_{\boldsymbol{\phi}}^{\text{ACW}} = \mathbf{g}_{\boldsymbol{\phi}}^{\text{base}} + \gamma_k \mathbf{g}_{\boldsymbol{\phi}}^{\text{sim}},$$

whereas the GRU-exclusive predictor parameters receive the full $\mathcal{L}_{\text{sim}}$ gradient throughout training. The shared final LayerNorm has both roles: its encoder-side contribution is attenuated, while its predictor-output contribution remains unscaled.

Writing p_k for the fraction of the 3,283 optimizer updates already completed, we use

$$\gamma(p_k) = \begin{cases} 0, & p_k < 0.02, \\ \dfrac{p_k - 0.02}{0.18}, & 0.02 \le p_k < 0.20, \\ 1, & p_k \ge 0.20. \end{cases}$$

Native and ACW are otherwise matched in architecture, initialization, data, batching, optimizer, learning-rate schedule, objective, training length, and evaluation protocol.

Training and evaluation protocol. Table 7 summarizes the settings used for all six formal Native/ACW runs.

Native and ACW within each seed are evaluated on the same 100 start–goal cases. For an episode, let e_t denote the mean squared error between the two-dimensional agent position and the goal position. The episode is successful when $e_t < 1$. For *Planning MSE*, we evaluate e_t at the first successful state when the goal is reached and at the final state otherwise, then average over the 100 episodes. The *final cross-wall rate* is the fraction of episodes whose final executed state lies on the goal side of the separating wall. The *mean termination step* averages the first successful state index, assigning unsuccessful episodes the value 201.

Table 8 reports the two quantities used in the main text together with two additional metrics produced by the same native evaluator. Values are mean $\pm$ sample standard deviation over seeds $101, 102, 103$.

Optimization trajectory. For the PLDM panel of Figure 4, we use the same $\mathcal{L}_{\text{sim}}$ optimized during training. Its instantaneous minibatch value is logged every 100 optimizer updates, producing 33 matched observations per run. For each of the three matched seed pairs we compute

$$\Delta(t) = \log_{10} \frac{\mathcal{L}_{\text{sim}}^{\text{ACW}}(t)}{\mathcal{L}_{\text{sim}}^{\text{Native}}(t)}.$$

No smoothing, interpolation, or checkpoint re-evaluation is applied.

Table 7: Training and closed-loop evaluation settings for PLDM.

Setting	Value
Training seeds	$101, 102, 103$
Training windows	4,350, horizon $H = 16$
Batch size	64
Optimizer	Adam, $\beta = (0.9, 0.999)$, weight decay 10^{-6}
Configured learning rate	5×10^{-4}
Effective peak learning rate	1.25×10^{-4}
Schedule	10% warm-up + cosine decay
Optimizer updates	3,283
Gradient clipping	none
Planner	MPPI
Candidate action sequences	500
MPPI noise / temperature	$\sigma = 5$, $\lambda = 0.005$
Planning horizon	up to 96
Replanning	every environment step
Action norm limit	2.45
Evaluation cases	100 cross-room cases / seed
Maximum episode length	200 actions

Table 8: Extended PLDM closed-loop planning evaluation. All metrics use the same formal evaluation protocol as the main-text results.

Metric	Native	ACW
Success rate ↑	0.470 ± 0.070	$\mathbf{0.560 \pm 0.044}$
Planning MSE ↓	178.87 ± 55.44	$\mathbf{143.72 \pm 48.25}$
Final cross-wall rate ↑	0.623 ± 0.080	$\mathbf{0.633 \pm 0.045}$
Mean termination step ↓	157.11 ± 3.42	$\mathbf{150.00 \pm 3.27}$

Reference-based mechanism diagnostic. The diagnostic in Figure 5 uses the same seeds $101, 102, 103$ as the formal Native–ACW comparison. Because the original formal runs did not retain sufficiently dense checkpoints for this analysis, we reconstruct the selected states using matched deterministic Native replays with the same training code, configuration, and seeds. The replay trajectories closely match their corresponding formal Native trajectories, although we do not treat the reconstructed states as bitwise-identical copies of the original runs.

At each audited training state, we freeze the complete representation parameters $\boldsymbol{\phi}$, including the shared final LayerNorm, and continue optimizing only the GRU-exclusive predictor parameters from their current values. The corresponding Adam state is restored, and the refitting objective is solely $\mathcal{L}_{\mathrm{sim}}$. Table 9 summarizes the diagnostic protocol.

The validation-best predictor is retained, so the resulting $\boldsymbol{\psi}_{\mathrm{ref}}$ is a finite better-fitted reference rather than an assumed exact optimum. Using the same fixed validation data for the current and reference predictors, we compute

$$\Delta_{\mathrm{obj}} = \mathcal{L}_{\mathrm{sim}}(\boldsymbol{\phi}, \boldsymbol{\psi}) - \mathcal{L}_{\mathrm{sim}}(\boldsymbol{\phi}, \boldsymbol{\psi}_{\mathrm{ref}}).$$

For the gradient comparison, all quantities are evaluated on the same fixed 256-sequence probe set. For each of four batches, we compute the raw $\mathcal{L}_{\mathrm{sim}}$ gradient with respect to all representation parameters; batch gradients are averaged, flattened, and concatenated without normalization. We then evaluate

$$\Delta^{m}_{\mathrm{grad}} = \left\| \mathbf{g}^{m}_{\boldsymbol{\phi}} - \mathbf{g}^{\mathrm{ref}}_{\boldsymbol{\phi}} \right\|_2, \qquad m \in \{\mathrm{Native}, \mathrm{ACW}\}.$$

At every audited progress, the ACW gradient uses the corresponding schedule value $\gamma(p)$. The current Native, current ACW, and better-fitted reference gradients therefore share the same representation state, probe data, and gradient construction.

Table 9: Protocol for the PLDM reference-based mechanism diagnostic.

Setting	Value
Seeds	101, 102, 103
Representation during refit	fixed
Predictor initialization	current predictor state
Predictor parameters updated	GRU-exclusive parameters only
Optimizer state	restored current Adam state
Reference objective	$\mathcal{L}_{\text{sim}}$
Fitting set	1,024 sequences
Validation set	512 sequences
Gradient-probe set	256 sequences
Batch size	64
Validation interval	100 predictor updates
Stopping tolerance	relative improvement $< 10^{-3}$
Patience	5 validation evaluations
Maximum refit length	50,000 updates
Gradient metric	raw Euclidean L_2 distance

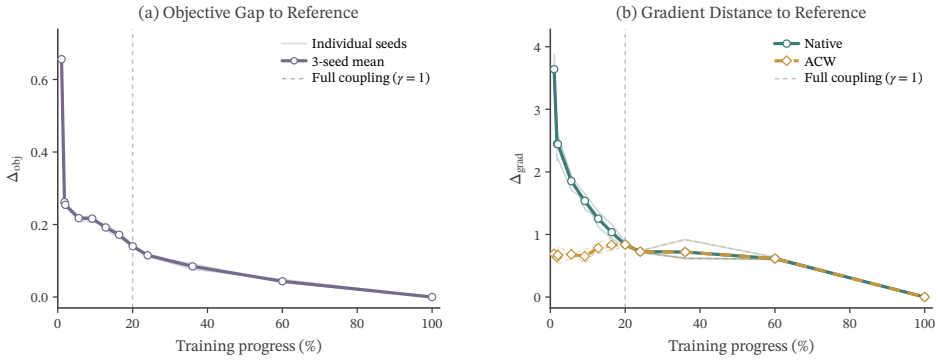


Figure 7: Full-range reference-based PLDM diagnostic over all 12 audited training progresses. **Left:** predictor-fitting objective gap between the current predictor and the finite better-fitted reference under a fixed representation. **Right:** Euclidean representation-gradient distance to the same reference under Native and ACW routing. Thin curves denote individual seeds and thick curves their three-seed mean. The vertical dashed line marks restoration of full coupling at 20% training progress ($\gamma = 1$); beyond this point, Native and ACW routing coincide. No smoothing or interpolation is applied.

Figure 7 extends the same quantities shown in the main-text diagnostic to all 12 audited training progresses, from approximately 1% through 100% of training. The predictor-fitting gap decreases over training, while during the attenuated-coupling phase the ACW-routed representation gradient is closer to the better-fitted reference. After full coupling is restored at 20% ($\gamma = 1$), Native and ACW gradient routing at the same replay state are identical and their distances to the reference consequently coincide.

C.3 STORM

Model architecture. We use the plain visual STORM baseline (Zhang et al., 2023) from the authors' newer OC-STORM codebase (Zhang et al., 2026); no object-centric features, segmentation model, or object-specific modules are used. Observations are resized to 64×64 RGB images and encoded into 32 categorical variables with 32 classes each. Table 10 summarizes the architecture used in our experiments.

Table 10: Architecture of the STORM model used in the MsPacman experiment. BN denotes batch normalization.

Component	Layer	#	Dimensions / setting	Activation / note
Encoder	Conv2d 4×4, stride 2	4	$3 \to 32 \to 64 \to 128 \to 256$	BN + SiLU
	Spatial resolution	–	$64 \to 32 \to 16 \to 8 \to 4$	–
Posterior	Linear	1	$4096 \to 32 \times 32$ logits	1% uniform mix
Transformer	Input stem	2	$(1024 + 9) \to 256 \to 256$	LayerNorm + ReLU
	Attention block	2	$d = 256$, 4 heads, FFN $= 512$	SiLU, dropout 0.1
	Context	–	maximum length 64	causal
Prior	Linear	1	$256 \to 32 \times 32$ logits	1% uniform mix
Decoder	Linear	1	$1024 \to 256 \times 4 \times 4$	BN + SiLU
	ConvTranspose2d 4×4, stride 2	4	$256 \to 128 \to 64 \to 32 \to 3$	BN + SiLU*
Reward head	Fully connected + output	3	$256 \to 256 \to 256 \to 255$	LayerNorm + ReLU
Termination head	Fully connected + output	3	$256 \to 256 \to 256 \to 1$	LayerNorm + ReLU
Actor	Fully connected + output	4	$1280 \to 512 \to 512 \to 512 \to 9$	LayerNorm + SiLU
Critic	Fully connected + output	4	$1280 \to 512 \to 512 \to 512 \to 255$	LayerNorm + SiLU

*BN and SiLU are applied to the hidden transposed-convolution layers; the output layer has no hidden activation.

Training setup. We train on `MsPacmanNoFrameskip-v4` under the Atari 100K protocol, using the minimal nine-action space and action repeat 4. The three independent training seeds are 0, 1, and 2, and Native and ACW runs are paired by seed. World-model updates use trajectory batches of 32×32 (batch size $\times$ sequence length) and Adam with learning rate 10^{-4}. After a replay warm-up of 1024 environment decisions, the configured model-training ratio is one world-model update per environment decision. Native and ACW use the same architecture, initialization seed, optimizer, replay protocol, loss weights, actor–critic training procedure, and environment interaction protocol.

Native gradient routing and ACW. STORM natively uses asymmetric gradient routing between its posterior and latent-dynamics prediction. Let

$$q_{t+1} = q_{\boldsymbol{\phi}}(z_{t+1} \mid o_{t+1})$$

denote the categorical posterior and let

$$p_{t+1} = p_{\boldsymbol{\psi}}(z_{t+1} \mid z_{\leq t}, a_{\leq t})$$

denote the next-latent distribution predicted by the Transformer dynamics model. Its two KL branches are

$$L_{\text{dyn}} = \max(1,\, D_{\text{KL}}[\text{sg}(q_{t+1}) \,\|\, p_{t+1}])\,, \tag{7}$$
$$L_{\text{rep}} = \max(1,\, D_{\text{KL}}[q_{t+1} \,\|\, \text{sg}(p_{t+1})])\,, \tag{8}$$

with native weights 0.5 and 0.1, respectively. Thus the weighted KL block is

$$L_{\text{KL}} = 0.5 L_{\text{dyn}} + 0.1 L_{\text{rep}}.$$

Importantly, stop-gradient on the target posterior in L_{dyn} does not eliminate its gradient to the representation: the source latent z_t is passed to the Transformer without detachment, so L_{dyn} can propagate through

$$q_{\boldsymbol{\phi}}(z_t \mid o_t) \to z_t \to \text{Transformer} \to p_{t+1}.$$

Accordingly, for STORM we instantiate the coupling component $\mathbf{g}_{\boldsymbol{\phi}}^{\text{dyn}}$ in ACW as the gradient contributed by the complete weighted KL block to the image encoder and posterior head. Denoting this contribution by $\mathbf{g}_{\boldsymbol{\phi}}^{\text{KL}}$, the implemented update is

$$\mathbf{g}_{\boldsymbol{\phi}}^{\text{ACW}} = \mathbf{g}_{\boldsymbol{\phi}}^{\text{Native}} - (1 - \gamma_k)\mathbf{g}_{\boldsymbol{\phi}}^{\text{KL}},$$

which directly matches the ACW update in the main text under the model-specific identification $\mathbf{g}_{\phi}^{\mathrm{dyn}} = \mathbf{g}_{\phi}^{\mathrm{KL}}$.

ACW modifies only this backward coupling: the forward KL values are unchanged, and the Transformer and prior head continue to receive their Native L_{dyn} gradients. Reconstruction, reward-prediction, termination-prediction, and actor–critic gradient pathways are also left unchanged. Letting ρ_k denote normalized progress in world-model optimizer updates, we use

$$\gamma_k = \begin{cases} 0, & \rho_k < 0.05, \\ \dfrac{\rho_k - 0.05}{0.15}, & 0.05 \le \rho_k < 0.20, \\ 1, & \rho_k \ge 0.20. \end{cases}$$

The KL-based dynamics-to-representation coupling is therefore suppressed during the first 5% of world-model optimization, linearly restored between 5% and 20%, and identical to Native training thereafter.

Evaluation and trajectory construction. For the greedy evaluation return in Table 2, we evaluate the final checkpoint produced by each training run, with no validation- or return-based checkpoint selection. Each seed is evaluated for 20 episodes using greedy action selection; STORM's native stochastic posterior sampling is retained. Returns are first averaged within each seed, and we report the mean and sample standard deviation across the three training seeds.

The late online return is the arithmetic mean of the final 20 completed training-episode returns in each run. These per-seed values are likewise summarized by their mean and sample standard deviation across seeds.

For the STORM trajectory in Figure 4, we monitor the raw posterior–prior mismatch

$$L_{\mathrm{pp}} = \mathbb{E}_{B,t}\left[\sum_{d=1}^{32} D_{\mathrm{KL}}(q_{t+1,d} \,\|\, p_{t+1,d})\right].$$

This quantity is taken before the free-bit lower bound, before the $0.5/0.1$ loss weights, and before ACW gradient scaling; hence it measures the posterior–prior mismatch itself rather than a quantity mechanically rescaled by γ_k.

The compact trajectory is recorded every 100 world-model optimizer updates and averaged within fixed 1% training-progress intervals, with no additional smoothing. Native and ACW runs are paired by seed, and the plotted quantity is

$$\Delta(t) = \log_{10} \frac{L_{\mathrm{pp}}^{\mathrm{ACW}}(t)}{L_{\mathrm{pp}}^{\mathrm{Native}}(t)}.$$