YOU ONLY LIVE ONCE: STOP THE GRADIENT OR REGULARIZE AND REPLAY?

Anonymous authors
Paper under double-blind review

ABSTRACT

An agent that learns a world model from its own experience sees one finite trajectory, streamed in time order. We study JEPA in this *you-only-live-once (YOLO)* setting and find that the standard stop-gradient target fails silently: with finite data, the encoder can lock onto spurious noise correlations while its latent variance and training loss look healthy. An analysis of the objective motivates this failure. Removing stop-gradient and adding a variance floor regularizer prevents the collapse, and replaying past experience widens the samples for estimating the variance. Our Replay Variance Floor (RVF) recovers the latent dynamics in controlled linear systems and improves planning over the stop-gradient baseline on a range of visual control tasks.

1 INTRODUCTION

An autonomous agent inhabits a temporal world, observing a stream of high-dimensional sensory data from which it must build an internal model of how the world evolves. Learning latent dynamics from sequential observations is a natural path to such world models. An encoder compresses observations into a compact latent space, and a simple predictor captures temporal transitions, enabling planning and control without operating in the observation space.

The Joint Embedding Predictive Architecture (JEPA) (LeCun, 2022) instantiates this idea. The JEPA objective alone admits a trivial solution where the encoder maps everything to a constant. A popular fix today in modern self-supervised learning (SSL) is the *stop-gradient* trick: pass the prediction target through the encoder but block gradients on that branch. This is used by many JEPA implementations (Assran et al., 2023; Bardes et al., 2024), and the closely related image SSL methods (Grill et al., 2020; Chen & He, 2021). Distributional regularizers (Zbontar et al., 2021; Bardes et al., 2022; Balestriero & LeCun, 2025) offer an alternative to stop-gradient. Nevertheless, all of these methods were developed for large image or video datasets, where each mini-batch draws fresh, independent samples from a large dataset.

Different from the traditional static setting, in this paper we study the *you-only-live-once* (YOLO) setting, where the agent adapts from a single, finite trajectory of its own experience. Whereas large-scale pretraining supplies robust features and long histories of experience, adaptation to a new environment must contend with both limited total experience and a local view of its history, just as for an embodied agent learning on the fly (Wang et al., 2026a).

We first demonstrate the problem in a linear dynamical system observed through high-dimensional noise. With stop-gradient, the encoder silently locks onto noise instead of signal, a failure we call *signal collapse*. Latent diagnostics such as variance, covariance, and the training loss itself all look healthy, making this phenomenon distinct from the well-known *variance collapse* problem.

We then analyze the encoder objective with the optimal predictor substituted in. Without stop-gradient, the target moves with the encoder, so the objective asks the whole latent to be predictable from its past. With stop-gradient and a fixed target, the objective only asks the encoder to predict the current latents, a one-sided problem on the trajectory's lag-covariance. With finite data, this lag-covariance contains spurious noise correlations that look like signal to the one-sided objective. Our controlled experiments show that stop-gradient can then lock the encoder onto noise.

To address the failure caused by finite experience, we remove stop-gradient and retain a variance floor along different directions. To address the local statistics imposed by streaming, replay keeps a small representative sample of the past so the floor can capture slow signals beyond the current short window. Our proposed method, the Replay Variance Floor (RVF), combines these fixes, recovers latent linear dynamics across the frequency spectrum, and outperforms existing regularizers.

Reviewers: please read the Reviewer Guidelines (iclr.cc/Conferences/2027/ReviewerGuidelines) and the AI Policy for Reviewers (iclr.cc/Conferences/2027/AIPolicyForReviewers). If you used AI to expand, edit, or polish your review, please provide the input text to the LLM. Better still, consider skipping the LLM and submitting your original text: we, and the authors, are much more interested in your unedited thoughts than in what an LLM has to say. AI-assisted or not, you are putting your name and reputation behind your review: LLM-generated falsehoods, hallucinations or misrepresentations are subject to disciplinary action, which may include desk-rejecting all papers you have authored.

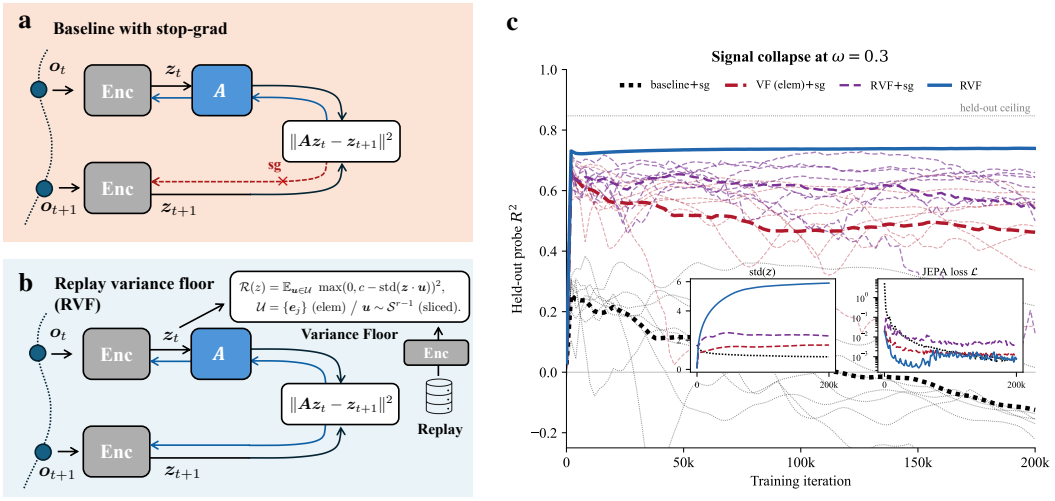


Figure 1: **Signal collapse in single-trajectory JEPA, and the Replay Variance Floor.** (**a**) JEPA with stop-gradient on the target branch. (**b**) RVF: a variance floor $\mathcal{R}(z)$ on the latents, estimated on the current window plus a replay buffer of past observations for better variance estimation. (**c**) Held-out probe over training on latent circular dynamics (ω=0.3, streaming). Insets: the JEPA training loss and the standard deviation $\mathrm{std}(z)$.

Beyond the synthetic curve experiments, we also evaluate realistic visual control environments spanning maze navigation, reaching, and manipulation. RVF consistently improves planning success over the stop-gradient baseline, and removing stop-gradient helps in most settings, including fine-tuning a pretrained world model under a visual shift.

2 Analysis

In this section, we analyze JEPA on a single trajectory of T observations under the assumption of a linear encoder and a linear predictor. The JEPA loss involves two learnable components: the encoder Φ and the predictor A. Because the predictor is linear, its optimum can be computed in closed form for any given encoder. Substituting this optimum back into the loss then eliminates A and leaves an objective that depends only on the encoder. The empirical sections test whether these insights transfer to nonlinear encoders.

2.1 Setting

JEPA. An encoder $\Phi : \mathbb{R}^d \to \mathbb{R}^r$ maps observations to latent states $\boldsymbol{z}_t = \Phi(\boldsymbol{o}_t)$. A linear predictor $A \in \mathbb{R}^{r\times r}$ models one-step transitions. For simplicity, we omit action conditioning, and the JEPA loss (LeCun, 2022) is:

$$\mathcal{L} = \frac{1}{T-1}\sum_{t=0}^{T-2} \left\| A\,\boldsymbol{z}_t \;-\; \boldsymbol{z}_{t+1} \right\|^2.$$

Without further modification, this loss admits the trivial solution $\Phi \equiv 0$. In practice, two mechanisms are used to prevent collapse: (i) a *stop-gradient* on the target $\boldsymbol{z}_{t+1}$, sometimes combined with an exponential-moving-average (EMA) (Assran et al., 2023; Bardes et al., 2024); or (ii) a *regularizer* on the latent distribution (Zbontar et al., 2021; Bardes et al., 2022; Balestriero & LeCun, 2025). As we show, these two choices lead to different encoder objectives. Throughout the main paper we use stop-gradient *without* EMA, which means the target is the current encoder with the gradient blocked. Adding EMA does not resolve signal collapse (Appendix C).

High-dimensional linear dynamics. For our analysis we use a theoretical generative model. A low-dimensional state $\boldsymbol{s}_t \in \mathbb{R}^k$ evolves under linear dynamics $\boldsymbol{s}_{t+1} = R\,\boldsymbol{s}_t$, where $R \in \mathbb{R}^{k\times k}$ is the transition matrix. The agent does not observe $\boldsymbol{s}_t$ directly. Instead, it receives high-dimensional observations generated by $\boldsymbol{s}_t$ in $\mathbb{R}^d$ via a random orthonormal map $U \in \mathbb{R}^{d\times k}$ and adding isotropic Gaussian noise:

$$\boldsymbol{o}_t = U\boldsymbol{s}_t + \sigma n_t, \qquad n_t \sim \mathcal{N}(0, I_d), \quad t = 0, 1, \ldots, T-1.$$

The noise level σ and trajectory length T govern the finite-sample uncertainty in the observations.

Spurious sample structure at finite T. Based on random-matrix theory, T samples of i.i.d. noise $n_t \in \mathbb{R}^d$, the empirical lag-1 cross-covariance $\frac{1}{T-1}\sum_t n_t n_{t+1}^\top$ has nonzero spectral norm even though its population value is zero. For $d \ll T$ its scale is of order $\sqrt{d/T}$, and it grows further for $d \gtrsim T$, so observation noise contributes spurious structure of order $\sigma^2\sqrt{d/T}$ to the observation lag-covariance.

2.2 Two JEPA Objectives

For a linear encoder $\Phi(o) = Wo$, define three sample statistics from the trajectory: the lag-0 covariance $\hat{\Sigma}_{00} = \frac{1}{T-1}\sum_t \boldsymbol{o}_t \boldsymbol{o}_t^\top$ (how observations co-vary with themselves), the next-step covariance $\hat{\Sigma}_{11} = \frac{1}{T-1}\sum_t \boldsymbol{o}_{t+1} \boldsymbol{o}_{t+1}^\top$, and the lag-1 cross-covariance $\hat{\Sigma}_{01} = \frac{1}{T-1}\sum_t \boldsymbol{o}_t \boldsymbol{o}_{t+1}^\top$ (how each observation co-varies with the next). We empirically mean-center observations before forming these statistics. In latent space they become $\Sigma_z = W\hat{\Sigma}_{00}W^\top$ and $C_z = W\hat{\Sigma}_{01}W^\top$, and we further assume Σ_z is invertible on the row space of W.

The key difference between the two settings is what serves as the prediction target. Without stop-gradient, $\boldsymbol{z}_{t+1} = W\boldsymbol{o}_{t+1}$ uses the same encoder as the prediction input. With stop-gradient, the target is $z'_{t+1} = W_{\mathrm{tgt}}\,\boldsymbol{o}_{t+1}$ where $W_{\mathrm{tgt}} = W$ but is treated as a constant when computing gradients.

Theorem 1 (Encoder objective after eliminating the predictor)**.** *For a linear encoder $\Phi(o) = Wo$, substituting the optimal linear predictor gives:*

1. ***Without stop-gradient****, $A^\star = C_z^\top \Sigma_z^{-1}$ and*

$$\mathcal{L}^\star_{\text{no-sg}}(W) \;=\; \mathrm{tr}\big(W\hat{\Sigma}_{11}W^\top\big) \;-\; \mathrm{tr}\big(C_z^\top \Sigma_z^{-1} C_z\big).$$

2. ***With stop-gradient and a frozen target*** *W_{tgt}, writing $\hat{\Sigma}_{zz'} = W\hat{\Sigma}_{01}W_{\mathrm{tgt}}^\top$, $A^\star = \hat{\Sigma}_{zz'}^\top \Sigma_z^{-1}$ and*

$$\mathcal{L}^\star_{\text{sg}}(W;\, W_{\mathrm{tgt}}) \;=\; \mathrm{tr}\big(W_{\mathrm{tgt}}\hat{\Sigma}_{11}W_{\mathrm{tgt}}^\top\big) \;-\; \mathrm{tr}\big(\hat{\Sigma}_{zz'}^\top \Sigma_z^{-1} \hat{\Sigma}_{zz'}\big).$$

Both objectives reward explained next-step variance, but they differ in whether the target variance depends on the encoder. In Part (1) the target moves with W, so the loss penalizes target variance that the predictor leaves unexplained, and up to target whitening the objective resembles CCA and the VAMP score (Wu & Noé, 2020). In Part (2) the target variance is constant, and the fixed-target optimum spans the directions that best predict the current latents, whether these carry signal or noise. The theorem characterizes the objectives. In training, the target follows the encoder after every update. It therefore motivates, rather than establishes, the hypothesis that stop-gradient locks onto spurious lag structure while training without stop-gradient favors signal. Section 4 tests this hypothesis and measures where the frequency cutoff falls, and Section A.3 discusses the limits of the analysis.

2.3 In-Sample Diagnostics Can Hide Signal Collapse

Proposition 2 (Latent covariance does not identify signal recovery)**.** *Let the faithful encoder $\Phi^\star$ project onto the signal directions U and the collapsed encoder $\tilde{\Phi}$ onto directions orthogonal to U, with unit signal covariance and isotropic noise of positive variance. Then the two population latent covariances differ only by a global scale, which a variance regularizer can hide. A linear probe of the true state separates them, with population R^2 equal to $1/(1+\sigma^2)$ at $\Phi^\star$ and 0 at $\tilde{\Phi}$.*

Latent variance, covariance, and in-sample lag correlation can look alike for faithful and collapsed encoders because variance regularization can hide scale differences and directions fitted to training noise can mimic the signal's lag correlation. Under time-independent noise, held-out temporal statistics can separate them because the collapsed encoder's lag structure does not carry over to a second trajectory, while held-out variance and covariance remain uninformative. We therefore measure state recovery directly with the held-out state probe of Section 4.1, with further details in Section A.4.

3 Regularize and Replay

In this section, we introduce complementary remedies for finite experience and for the local statistics imposed by streaming. A variance floor (VF) without stop-gradient addresses the finite-sample failure, which is most severe for fast signals, whereas replay widens the sample used to estimate variance so the encoder can capture slow signals. Our Replay Variance Floor (RVF) combines the two strategies.

Anti-collapse regularizers. Regularizers prevent constant representations by constraining their distribution. VICReg (Bardes et al., 2022) combines element-wise variance and covariance penalties, while SIGReg, the sketched isotropic Gaussian regularizer (Balestriero & LeCun, 2025), matches Gaussianity along random slices. These regularizers were designed to prevent scale or shape collapse in i.i.d. multi-view batches with well-estimated statistics, whereas here every statistic comes from one short, correlated trajectory, so a method must withstand estimation error. To this end, we adapt the variance floor idea to this setting explained below.

Variance floors. Unlike SIGReg, a variance floor penalizes insufficient variation without matching a full distribution. We use a one-sided squared hinge on standard deviations below a threshold.

- *Element-wise.* The element-wise floor, a squared form of the variance term of VICReg (Bardes et al., 2022), prevents each coordinate from losing variance:

$$\mathcal{R}_{\text{elem}}(z) = \frac{1}{r}\sum_{i=1}^{r} \text{relu}(1 - \text{std}_t(z_{t,i}))^2, \tag{1}$$

- *Sliced.* The sliced floor constrains variance along random directions, penalizing flat directions along correlated coordinates:

$$\mathcal{R}_{\text{slice}}(z) = \mathbb{E}_{\boldsymbol{u}}\Big[\text{relu}(1 - \text{std}_t(\boldsymbol{z}_t \cdot \boldsymbol{u}))^2\Big], \qquad \boldsymbol{u} \sim \text{Unif}(\mathbb{S}^{r-1}), \tag{2}$$

  where $\boldsymbol{z}_t \cdot \boldsymbol{u}$ projects each latent onto the direction $\boldsymbol{u}$ and std_t is taken over the estimation sample, which replay widens below.

A floor is appropriate because it remains inactive as long as the standard deviations reach the threshold. In contrast, a two-sided variance penalty continues to squeeze anisotropic covariance.

Replay. Replay broadens the sample used to estimate variance beyond the current window. Let $\mathcal{B}_t$ hold M past observations after the buffer has filled, and let $\mathcal{W}_t$ denote the current observation window. At each step, the regularizer uses $\boldsymbol{z} = \Phi(\mathcal{W}_t \cup \mathcal{B}_t)$, re-encoding stored observations. Replay can also enter the JEPA objective by storing windows of frames. We find it useful in streaming visual control environments.

In practice, we consider the following replay strategies.

- **First-in, first-out (FIFO).** FIFO replay stores the M most recent observations in a ring: $\mathcal{B}_t = (o_{t-M+1}, \ldots, \boldsymbol{o}_t)$ once full, replacing the oldest entry on arrival (a ring buffer (Chaudhry et al., 2019)). Its temporal span is M.
- **Reservoir.** Reservoir replay keeps a small, representative sample of everything seen. It stores the first M observations, and for each arrival $\boldsymbol{o}_t$ with $t > M$, it replaces a uniformly chosen slot with probability M/t (Vitter, 1985).
- **Constant turnover (CT).** CT replay keeps refreshing at a fixed rate. Each frame replaces a uniformly chosen buffer slot with fixed probability p, independent of stream position. Slot lifetimes are geometric with mean M/p frames, making the buffer an exponentially weighted sample of the recent stream. Once full, this is the replacement rule of exponentially biased reservoir sampling (Aggarwal, 2006).

The Replay Variance Floor (RVF). By default, RVF uses the sliced variance floor $\mathcal{R}_{\text{slice}}$, estimated on the current window together with replayed observations, trained without stop-gradient. The replay buffer uses the CT variant by default, with FIFO and reservoir replay as variants for ablation.

4 Experiment 1: Latent Linear Dynamics

To illustrate the limitation of JEPA learning with stop gradient and other regularizers, in Experiment 1, we first use latent linear dynamics that match our analysis. Realistic visual control experiments are reported in Experiment 2.

4.1 Setup

We use the linear dynamics model in Section 2.1 to generate latent curves with high-dimensional noise. The main experiments use circles at σ=0.3 over T=500 frames with varying frequencies, and Table 3 extends the evaluation to helix, lissajous, 3-freq, cone, and spiral curves at σ=0.1. We primarily test the more realistic learning scenario of streaming windows, with i.i.d. sampling as a control in Table 9. Implementation details are included in Appendix B.

sg	floor	replay	ω=0.01	ω=0.05	ω=0.1	ω=0.3	ω=1
			held-out probe R^2 / ten-step rollout R^2				
$T = 500$							
oracle ceiling (R^2)			*0.83*	*0.84*	*0.84*	*0.84*	*0.84*
✓	–	–	0.19 / div.	0.28 / div.	0.28 / div.	-0.13 / div.	-0.02 / div.
✓	elem	–	0.41 / -0.08	0.51 / 0.48	0.60 / 0.51	0.53 / 0.34	0.30 / div.
✓	slice	–	0.44 / 0.30	0.58 / 0.56	0.61 / 0.59	0.60 / 0.57	0.54 / 0.56
✓	elem	FIFO	0.51 / 0.31	0.55 / 0.51	0.57 / 0.55	0.45 / 0.11	0.21 / div.
✓	slice	FIFO	0.56 / **0.57**	0.60 / 0.57	0.59 / 0.61	0.59 / 0.61	0.58 / 0.55
–	elem	–	0.08 / -0.11	0.06 / -0.21	0.48 / div.	0.73 / -0.43	**0.74** / 0.51
–	slice	–	0.01 / div.	0.02 / div.	0.73 / **0.72**	**0.75** / **0.73**	**0.74** / **0.73**
–	elem	FIFO	-0.10 / div.	0.50 / div.	0.65 / div.	0.69 / div.	0.69 / div.
–	slice	FIFO	0.10 / div.	**0.74** / 0.70	**0.74** / 0.71	**0.75** / **0.73**	0.73 / 0.71
–	elem	CT	0.24 / div.	0.48 / div.	0.62 / div.	0.68 / div.	0.69 / div.
–	slice	CT	**0.69** / 0.50	**0.74** / **0.71**	0.73 / 0.71	0.73 / 0.69	0.72 / 0.71

Table 1: **Main results on circular latent dynamics** ($T = 500, \sigma = 0.3$). Appendix Table 7 reports T=5,000 with reservoir replay.

Evaluation protocol. The held-out probe measures whether the learned representation preserves the signal on a fresh episode. Observation noise limits the held-out probe to an oracle ceiling. Additionally, we also test held-out R^2 on ten-step forward rollouts, which measures the quality of the predictor.

Compared methods. We compare the unregularized JEPA baseline with stop-gradient (baseline+sg) against the element-wise and sliced variance floors, with and without stop-gradient and replay in Table 1. We then compare alternative regularizer forms under identical replay in Table 2. **SIGReg** matches distributional shape along random slices. **VICReg** pairs the element-wise floor with an off-diagonal covariance penalty. The two-sided **variance penalty (VP)** penalizes $(\mathrm{std}_t(z) - 1)^2$ in both directions, so it also punishes excess variance. The **log-det floor** penalizes $\sum_j \mathrm{relu}(-\log \mu_j)$ over covariance eigenvalues μ_j. We swept each variant's coefficient λ throughout, and report the best performance.

4.2 Results

Main result. Figure 2a,b shows the stop-gradient failure: baseline+sg fails across the frequency spectrum and recovers only slow signals once experience reaches thousands of frames. Table 1 shows that stop-gradient performs well below the oracle ceiling, even with a variance floor and replay. Without stop-gradient, the sliced floor performs well at fast frequencies but fails at slow ones. Replay restores the slow frequencies, with CT reaching further than FIFO. The element-wise floor reaches comparable probe values with replay, but its rollouts diverge, while the sliced floor rolls out stably. Only RVF, the sliced floor with CT replay and no stop-gradient, recovers the signal across the whole spectrum with stable rollouts, and Figure 2c shows that this holds across trajectory lengths.

Sample complexity. Under streaming learning, unregularized stop-gradient needs 10–25$\times$ more experience than RVF where it recovers at all, and never recovers at the fastest frequencies within the measured budget (Figure 2b,c). RVF stands out with limited experience, whereas with abundant data, the element-wise variance floor with stop-gradient is comparable (Appendix Table 10).

Regularizer ablations. Table 2 compares regularizer forms under the same CT replay strategy. One-sided VF wins when data is scarce, because the two-sided VP is too strong for limited data, while the shape-matching SIGReg underperforms, likely due to the difficulty of estimating covariance. With ten times the data, the alternative forms close the gap, and the log-det floor is comparable with RVF (Table 8). The rollout R^2 shows greater differences among regularizers: the two-sided penalty's predictor diverges throughout, SIGReg's degrades at fast frequencies, and the log-det floor, VICReg, and RVF roll out stably. Slicing outperforms the element-wise variant, especially with limited data, and retains an advantage with more data (also see the Appendix Table 9 for i.i.d. results).

Replay strategies. The variance cannot be properly measured if the signal is moving too slowly in the current window (Figure 2a). Replay is employed to improve slow-frequency recovery while preserving the fast-frequency gains (Table 1). FIFO reaches its span limit at the slowest frequency, and CT removes this limit while matching FIFO elsewhere (Table 1). Reservoir replay also recovers the slow band at small T but freezes and fails on long streams (Appendix Figure 4; Table 7). Replaying into the JEPA objective as well does not help here (see Appendix Table 6).

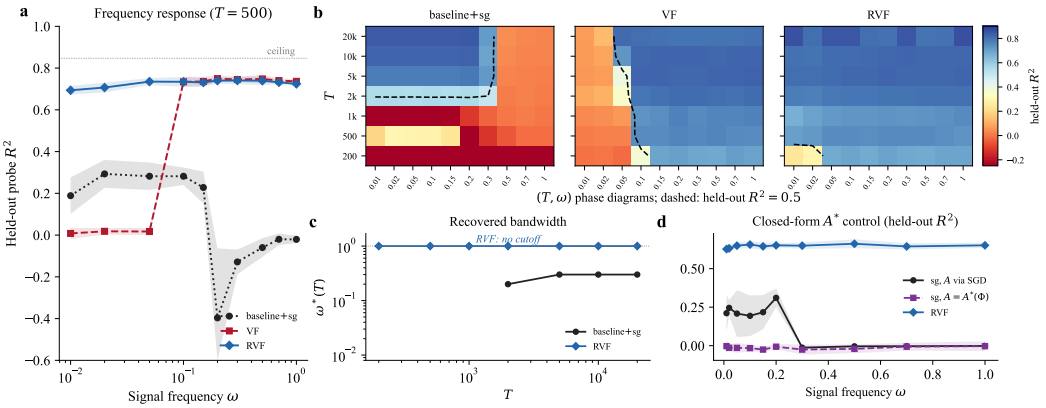


Figure 2: **Frequency response of stop-gradient and RVF** (circle, streaming, σ=0.3, 8 seeds). (**a**) Frequency response at T=500. (**b**) Phase diagrams comparing baseline+sg, VF, and RVF. (**c**) RVF succeeds at a wide range of sequence lengths. (**d**) The closed-form A^* does not rescue stop-gradient.

form	ω=0.01	ω=0.05	ω=0.1	ω=0.3	ω=1
	held-out probe R^2 / ten-step rollout R^2				
$T = 500$					
oracle ceiling (R^2)	*0.83*	*0.84*	*0.84*	*0.84*	*0.84*
SIGReg	0.28 / div.	0.45 / -0.16	0.53 / -0.39	0.57 / 0.45	0.57 / 0.56
VICReg	0.54 / 0.52	0.59 / 0.58	0.58 / 0.57	0.61 / 0.57	0.62 / 0.63
VP (two-sided)	-1.76 / div.	-0.83 / div.	0.48 / div.	0.63 / div.	0.52 / -0.65
log-det floor	0.63 / **0.62**	0.66 / 0.66	0.66 / 0.67	0.66 / 0.66	0.64 / 0.65
RVF (elem)	0.24 / div.	0.48 / div.	0.62 / div.	0.68 / div.	0.69 / div.
RVF	**0.69** / 0.50	**0.74** / **0.71**	**0.73** / **0.71**	**0.73** / **0.69**	**0.72** / **0.71**

Table 2: **Regularizer comparison in circular latent dynamics** (with CT replay). Appendix Table 8 reports the comparison at T=5,000.

More linear dynamic environments. In addition to circles, we also evaluate across helix, lissajous, 3-freq, cone, and spiral in Table 3, and observe consistent findings. The unregularized baseline+sg fails in fast curves while RVF stays near oracle ceiling.

sg	floor	replay	helix	lissajous	3-freq	cone	spiral
			held-out R^2, slow ω=0.1 / fast ω=1.0				
oracle ceiling (R^2)			0.98 / 0.98	0.98 / 0.98	0.98 / 0.98	1.00 / 1.00	0.95 / 0.95
✓	–	–	0.91 / -0.02	0.42 / -0.03	0.76 / -0.09	**0.98** / -0.02	0.78 / -0.02
✓	elem	–	0.95 / 0.96	0.83 / 0.91	0.86 / 0.89	**0.98** / 0.97	0.84 / 0.81
–	elem	FIFO	0.93 / 0.94	0.84 / 0.77	0.78 / 0.82	**0.98** / **0.98**	0.86 / 0.90
–	slice	–	0.77 / 0.03	0.94 / **0.96**	0.93 / **0.90**	0.96 / 0.94	0.89 / 0.88
–	slice	FIFO	**0.98** / **0.98**	**0.96** / 0.85	0.94 / 0.88	0.96 / 0.95	0.90 / 0.90
–	slice	CT	**0.98** / **0.98**	**0.96** / 0.85	**0.95** / 0.89	0.97 / 0.94	**0.91** / **0.91**

Table 3: Results on curves under streaming at T=500, σ=0.1. Appendix Table 11 reports the comparison at T=5,000.

5 Experiment 2: Latent Dynamics for Visual Control

We extend Section 4 to visual goal-reaching in the YOLO setting with an action-conditioned JEPA, on three control environments and on fine-tuning a pretrained world model.

5.1 Setup

Environments. We evaluate the PointMaze UMaze and Medium mazes (Fu et al., 2020), Reacher-Hard from the DeepMind Control Suite (Tassa et al., 2018), OGBench-Cube (Park et al., 2025), and

			U-maze		Medium maze		Reacher		Cube	
sg	regularizer	replay	i.i.d.	stream	i.i.d.	stream	i.i.d.	stream	i.i.d.	stream
✓	–	–	7.8 ±3.7	21.3 ±11.8	5.1 ±3.4	11.8 ±2.3	70.4 ±33.3	29.1 ±4.0	14.6 ±4.2	11.4 ±2.0
–	VICReg	–	69.8 ±5.4	52.2 ±5.0	51.1 ±7.8	25.6 ±4.4	46.9 ±2.2	42.9 ±1.4	14.9 ±1.1	13.6 ±2.5
–	SIGReg	–	70.4 ±5.7	39.3 ±22.3	40.9 ±31.0	14.2 ±3.8	59.1 ±6.0	40.7 ±2.7	13.6 ±1.0	**14.1** ±2.0
✓	elem	–	56.9 ±16.4	42.0 ±7.2	62.2 ±15.8	28.0 ±7.9	86.3 ±3.1	35.3 ±1.9	10.1 ±1.2	10.3 ±0.6
–	elem	–	88.7 ±3.5	37.6 ±5.4	61.1 ±11.9	24.4 ±2.3	61.8 ±2.2	39.9 ±6.1	14.9 ±1.9	13.4 ±1.6
✓	slice	–	58.0 ±0.7	51.1 ±1.4	45.8 ±3.4	31.8 ±13.4	85.6 ±3.6	86.4 ±3.2	11.2 ±1.6	13.8 ±0.4
–	slice	–	<u>96.0</u> ±2.4	57.3 ±6.4	**75.8** ±6.5	36.9 ±4.4	<u>88.0</u> ±0.0	43.0 ±0.9	15.7 ±1.3	12.7 ±1.8
✓	slice	CT	52.0 ±5.8	59.1 ±5.1	58.9 ±4.1	42.2 ±7.6	79.9 ±4.8	**89.6** ±0.7	10.1 ±0.4	10.6 ±1.6
–	slice	CT	95.1 ±2.5	<u>94.9</u> ±3.1	<u>74.7</u> ±2.4	**73.3** ±4.8	**89.7** ±0.6	73.3 ±11.9	**17.3** ±1.9	<u>13.9</u> ±0.8
✓	slice	CT + JEPA	57.1 ±0.8	54.2 ±3.8	61.6 ±12.5	40.9 ±6.8	67.9 ±4.5	82.9 ±2.0	11.3 ±0.7	11.6 ±1.0
–	slice	CT + JEPA	**96.9** ±2.7	**97.3** ±0.7	73.1 ±1.4	<u>72.7</u> ±4.4	85.4 ±0.7	<u>87.0</u> ±2.0	<u>16.4</u> ±1.0	<u>13.9</u> ±0.4

Table 4: **Planning success (%) on JEPA learning for visual control.** Frozen DINOv2-S encoder with a trained projector, MPC success after 20 replans, mean ± s.d. over 3 seeds. Bold: best mean per column; underline: second best.

PushT (Chi et al., 2025) with various visual shifts (Wang et al., 2026a). Whereas prior visual world models learn from many trajectories (Zhou et al., 2025), we train on one trajectory per environment and use a second independent trajectory for validation, PointMaze and Reacher trajectories are collected with a uniform-random action policy, and Cube trajectories with the OGBench scripted expert policy (Park et al., 2025). Each trajectory contains $T = 5{,}000$ observed frames, sampled every $K = 5$ simulator steps. We sample the goals in both mazes to ensure the goal distance is at least 3 maze units. For the rest environments, the goal is sampled 25 steps away from test data, and we additionally require the cube to move at least 4 cm to rule out trivial goals.[1]

Model. A frozen DINOv2 ViT-S/14 supplies patch tokens to a trainable convolutional global projector, which produces the latent representation used for prediction and planning. The projector has three stride-2 convolutional layers followed by pooling and outputs a 128-dimensional latent with LayerNorm. A transformer predictor embeds actions and concatenates them with the latents, using a causal temporal mask and a three-frame history to predict the next latent. Freezing the backbone fixes the features $\phi(\boldsymbol{o})$ that the projector reads, so with a linear projector and predictor, Theorem 1 would hold exactly with the sample covariances computed over $\phi(\boldsymbol{o})$ (Section A.3). Our projector and predictor are nonlinear, so these experiments test whether the mechanism carries over beyond the theorem.

Training. We compare i.i.d. and stream sampling over the same fixed training trajectory. Each update uses 32 four-frame windows, comprising three history frames and one target. The i.i.d. sampler draws windows at random positions, whereas the stream sampler takes consecutive, overlapping windows. We train for 10k steps on PointMaze and 50k on Reacher and Cube. We also test the sliced variance floor with CT replay (Section 3), and variants whose stored windows additionally enter the JEPA prediction loss, denoted "CT + JEPA" in Table 4. The CT buffer holds 128 frame slots on PointMaze and 512 on Reacher and Cube. We sweep $\lambda \in \{1, 10, 100\}$ and report the best performer per cell.

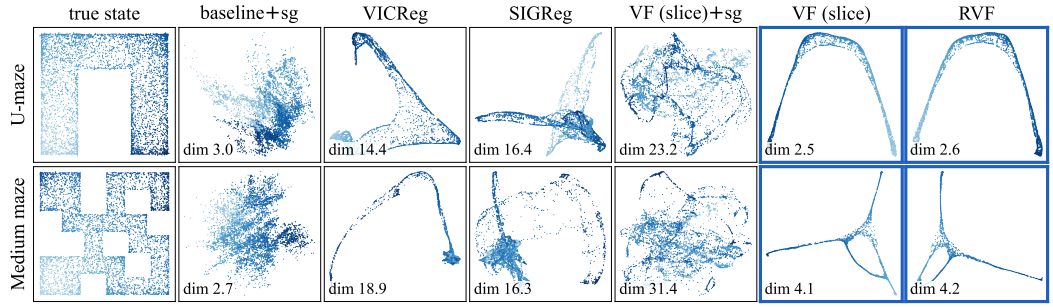
Planning. We evaluate goal-conditioned model predictive control (MPC), which minimizes the predicted latent distance to a goal image's latent. On PointMaze and Reacher, a gradient-descent planner uses 100 optimization steps and a horizon of 50 environment steps, executing 5 environment steps before replanning for 20 replans, over 50 start-goal pairs on PointMaze and 100 on Reacher. On Cube, whose goals follow the replay-goal construction of LeWorldModel (Maes et al., 2026), a CEM planner (300 samples, 10 iterations) replans every 5 environment steps for 20 replans over 100 pairs. For each of 3 training seeds, we average over 3 planning seeds and report the mean and standard deviation over training seeds. Additional planning hyperparameters appear in Appendix E.

Main Results. Table 4 shows our results on planning for visual control. On Reacher, its i.i.d. spread reflects bimodal seeds. The sliced VF matches or exceeds the element-wise VF in all but two matched cells, both within noise, so we focus on the sliced VF here. As in Section 4, stream sampling without replay hurts the sliced VF, from 96.0% to 57.3% on PointMaze and from 88.0% to 43.0% on Reacher, since each update sees a short, correlated stretch of the trajectory. Replay decorrelates this sample, with RVF recovering 94.9% and 73.3%. Replaying into the prediction loss as well reaches 97.3% and 87.0%, though it brings no consistent gain under i.i.d. sampling. The medium maze shows the same

[1]This filtering contributes to why overall our Cube success rates are lower than those of related works (Maes et al., 2026), where start-goal pairs are not filtered.

Table 5: **Fine-tuning a pretrained PushT world model under a visual shift.** Planning success (%) after one epoch over N=16 target-domain trajectories (best λ per cell).

			Red agent		Red block		Blur	
sg	regularizer	replay	i.i.d.	stream	i.i.d.	stream	i.i.d.	stream
✓	–	–	58.0 ±2.0	42.0 ±3.5	24.7 ±3.1	17.3 ±1.2	54.7 ±8.1	44.7 ±8.1
–	VICReg	–	51.3 ±1.2	16.7 ±3.1	10.0 ±0.0	6.0 ±2.0	47.3 ±7.6	21.3 ±3.1
–	SIGReg	–	60.0 ±2.0	48.7 ±9.2	57.3 ±11.0	42.0 ±10.0	60.7 ±4.2	50.7 ±2.3
✓	elem	–	56.7 ±3.1	45.3 ±6.1	50.0 ±5.3	40.0 ±2.0	58.0 ±2.0	44.7 ±6.1
–	elem	–	62.7 ±8.1	46.0 ±0.0	58.7 ±9.0	42.7 ±1.2	60.7 ±7.0	46.7 ±6.1
✓	slice	–	56.7 ±5.0	55.3 ±10.1	53.3 ±8.1	36.0 ±3.5	61.3 ±3.1	45.3 ±8.1
–	slice	–	**63.3** ±1.2	44.0 ±3.5	59.3 ±12.1	42.7 ±2.3	59.3 ±5.0	48.7 ±1.2
✓	slice	CT	58.7 ±3.1	44.0 ±4.0	52.0 ±7.2	36.0 ±4.0	61.3 ±2.3	46.7 ±3.1
–	slice	CT	62.7 ±1.2	43.3 ±4.2	**60.7** ±13.3	48.0 ±7.2	58.0 ±4.0	46.0 ±7.2
✓	slice	CT + JEPA	57.3 ±7.0	46.0 ±5.3	46.0 ±5.3	43.3 ±7.6	60.0 ±2.0	51.3 ±8.1
–	slice	CT + JEPA	60.7 ±4.2	**56.7** ±4.6	56.0 ±8.7	**52.0** ±5.3	**62.7** ±5.0	**58.7** ±6.1



	U-maze				Medium maze			
method	dim	ρ all	ρ far	straight.	dim	ρ all	ρ far	straight.
true state	*2*	–	–	*0.41*	*2*	–	–	*0.44*
baseline+sg	3.0 ±0.7	0.44 ±0.09	0.09 ±0.04	-0.32 ±0.04	2.7 ±0.9	0.49 ±0.05	0.20 ±0.01	-0.37 ±0.05
VICReg	14.4 ±0.2	0.61 ±0.01	0.39 ±0.01	-0.02 ±0.01	18.9 ±0.7	0.49 ±0.04	0.33 ±0.05	0.27 ±0.01
SIGReg	16.4 ±0.9	0.45 ±0.04	0.23 ±0.05	-0.11 ±0.02	16.3 ±0.7	0.42 ±0.04	0.25 ±0.06	0.22 ±0.00
VF (slice)+sg	23.2 ±0.6	0.12 ±0.04	-0.04 ±0.05	-0.29 ±0.01	31.4 ±1.1	0.10 ±0.03	0.00 ±0.01	-0.09 ±0.01
VF (slice)	2.5 ±0.0	0.98 ±0.00	0.92 ±0.01	0.28 ±0.00	4.1 ±0.1	0.94 ±0.02	0.73 ±0.09	0.39 ±0.00
RVF	2.6 ±0.1	0.98 ±0.00	0.92 ±0.01	0.28 ±0.00	4.2 ±0.2	0.93 ±0.01	0.73 ±0.04	0.40 ±0.01

Figure 3: **Latent geometry on PointMaze.** Top: PCA of held-out latents on the U-maze and a larger maze with loops, with dim the effective dimension. Bottom: Spearman correlation ρ between Euclidean latent distance and distance along the maze, over all pairs and the farther half, and trajectory straightness (Wang et al., 2026b); mean ± s.d. over 3 seeds.

pattern. Dropping stop-gradient raises the sliced VF's mean success in all matched i.i.d. comparisons and in most matched stream comparisons with replay, so it is the better choice whenever the variance statistic has a larger sample. The exception is Reacher under stream sampling, where the sliced VF with stop-gradient keeps its i.i.d. performance and, with CT replay, gives the best stream result.

Other regularizers. On PointMaze and Reacher, VICReg and SIGReg fall well below RVF under both i.i.d. and stream sampling (Table 4). All regularized latents decode position, but only the variance floors without stop-gradient embed it in a few dimensions where Euclidean latent distance, which the planner minimizes, tracks distance in the maze (Figure 3). On Cube, RVF is best on i.i.d. windows, while under stream sampling SIGReg matches the sliced variants. SIGReg acts on the raw projector output with the output LayerNorm removed.

Transfer a pretrained model to a new test environment. We fine-tune the released AdaJEPA world model (Wang et al., 2026a), pretrained on in-distribution PushT, on three visual shifts: red agent, red block, and gaussian blur. Each method supplies the fine-tuning objective for a single pass over $N = 16$ shifted trajectories (each with an average length of 140 simulator steps), the

practical setting of adapting to a test domain from little data. We compare i.i.d. and stream sampling, using one fine-tuning seed and the same λ sweep as above, with the best performer reported per cell. Fine-tuning with stop-gradient alone gives limited gains over the unadapted model and falls below it under stream sampling (Table 5). Variance floors without stop-gradient give the strongest i.i.d. results, and under streaming the replay of past windows into the JEPA loss helps, giving the best stream cell on every shift.

6 Related Work

Self-supervised learning and anti-collapse. Modern SSL methods learn representations by aligning different views of the same input. Contrastive methods such as SimCLR (Chen et al., 2020) and MoCo (He et al., 2020) use negative pairs to prevent collapse. Non-contrastive alternatives avoid negatives: BYOL (Grill et al., 2020) and SimSiam (Chen & He, 2021) use a stop-gradient on the target branch (with or without EMA), while DINO (Caron et al., 2021) combines stop-gradient with centering. Garrido et al. (2023) show that contrastive and non-contrastive objectives are algebraically related and equivalent under limited assumptions. Regularization-based approaches include Barlow Twins (Zbontar et al., 2021) (decorrelation), VICReg (Bardes et al., 2022) (element-wise standard-deviation floor), SIGReg (Balestriero & LeCun, 2025), and VISReg (Wu et al., 2026), the latter two matching a Gaussian *shape* along random one-dimensional slices. Our variance floor is one-sided like VICReg's and sliced like SIGReg's.

JEPA and world models. JEPA (LeCun, 2022) learns latent world models via temporal prediction, and I-JEPA (Assran et al., 2023) and V-JEPA (Bardes et al., 2024) scale it to images and video. Latent dynamics models are widely used for planning and control: world models (Ha & Schmidhuber, 2018), PlaNet (Hafner et al., 2019), Dreamer (Hafner et al., 2020; 2021), TD-MPC (Hansen et al., 2022; 2024), and DINO-WM (Zhou et al., 2025) all learn dynamics in latent space. Terver et al. (2025) study what drives planning success in JEPA world models, while Wang et al. (2026b) use explicit curvature regularization for latent planning. Closest to our setting, AdaJEPA (Wang et al., 2026a) adapts a pretrained JEPA world model at test time inside the planning loop, learning from its recent transitions under stop-gradient, where the restricted online update, one gradient step per replan, limits collapse.

Temporal representation learning. Slow Feature Analysis (Wiskott & Sejnowski, 2002) extracts slowly varying features under unit-variance and decorrelation constraints, and CPC (van den Oord et al., 2018) contrasts future observations against unrelated ones. Koopman-based methods instead learn representations in which the dynamics act linearly: VAMP (Wu & Noé, 2020) scores them by the whitened lagged covariance they explain, Lusch et al. (2018) learn them with deep autoencoders, and Ruiz-Morales et al. (2025) relate JEPA to Koopman invariant subspaces. Our analysis shares this linear setting, where the no-stop-gradient JEPA objective is VAMP-like (Theorem 1), but RVF places no constraint on the predictor and applies to the nonlinear models of our visual experiments.

7 Conclusion

Stop-gradient was built for standard JEPA when its statistics are well estimated, as with a large number of i.i.d. batches or a slow signal and a long history, such as paired image augmentations or offline dynamical transitions. However, embodied agents live through one trajectory and one task at a time where data scarcity and non-stationarity dominate. During single-trajectory learning, stop-gradient quietly fails and lets the encoder chase noise instead of the true signal. The remedy is to regularize rather than freeze one side of the objective, and to widen the sample through replay. In latent linear dynamics, RVF recovers the signal across the frequency spectrum. In visual control and when fine-tuning a pretrained world model, removing stop-gradient helps in most settings.

Stop the gradient, or regularize and replay? In the YOLO setting, regularize and replay.

Limitations. Our work has three main limitations. First, we study agents that adapt from a single finite trajectory, when pretraining has not already captured the dynamics of the environment, and a strong enough frozen representation could avoid the failure, although frozen DINOv2 features do not. Second, although the theory covers linear encoders and predictors, the same failure appears empirically with linear, tanh, and ReLU encoders, and we do not have a nonlinear theorem. Third, the proposed regularizers are found to be effective in the YOLO setting, but they have not been tested in offline invariance learning (e.g. Image SSL).

AI USE STATEMENT

LLM-based assistants (Claude, Codex) were used throughout this work under authors' direction and review, for developing the theoretical claims, implementing experiments, launching experiments, aggregating results, drafting the paper, verifying bibliography entries, proposing candidate interpretations of results. The research questions, the overall experimental design, the schematic figures, and all final interpretations and claims are the authors' own. The authors reviewed all AI-assisted content and take full responsibility for the contents of this paper.

REFERENCES

Charu C. Aggarwal. On biased reservoir sampling in the presence of stream evolution. In *Proceedings of the 32nd International Conference on Very Large Data Bases (VLDB)*, pp. 607–618, 2006.

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. In *CVPR*, 2023.

Randall Balestriero and Yann LeCun. LeJEPA: Provable and scalable self-supervised learning without the heuristics. *arXiv preprint arXiv:2511.08544*, 2025.

Adrien Bardes, Jean Ponce, and Yann LeCun. VICReg: Variance-invariance-covariance regularization for self-supervised learning. In *ICLR*, 2022.

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. *arXiv preprint arXiv:2404.08471*, 2024.

Mathilde Caron, Hugo Touvron, Ishan Misra, Hervé Jégou, Julien Mairal, Piotr Bojanowski, and Armand Joulin. Emerging properties in self-supervised vision transformers. In *ICCV*, 2021.

Arslan Chaudhry, Marcus Rohrbach, Mohamed Elhoseiny, Thalaiyasingam Ajanthan, Puneet K. Dokania, Philip H. S. Torr, and Marc'Aurelio Ranzato. On tiny episodic memories in continual learning. *arXiv preprint arXiv:1902.10486*, 2019.

Ting Chen, Simon Kornblith, Mohammad Norouzi, and Geoffrey Hinton. A simple framework for contrastive learning of visual representations. In *ICML*, 2020.

Xinlei Chen and Kaiming He. Exploring simple siamese representation learning. In *CVPR*, 2021.

Cheng Chi, Zhenjia Xu, Siyuan Feng, Eric Cousineau, Yilun Du, Benjamin Burchfiel, Russ Tedrake, and Shuran Song. Diffusion policy: Visuomotor policy learning via action diffusion. *The International Journal of Robotics Research*, 44(10-11):1684–1704, 2025.

Justin Fu, Aviral Kumar, Ofir Nachum, George Tucker, and Sergey Levine. D4rl: Datasets for deep data-driven reinforcement learning, 2020.

Quentin Garrido, Yubei Chen, Adrien Bardes, Laurent Najman, and Yann LeCun. On the duality between contrastive and non-contrastive self-supervised learning. In *ICLR*, 2023.

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. In *NeurIPS*, 2020.

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

Danijar Hafner, Timothy Lillicrap, Ian Fischer, Ruben Villegas, David Ha, Honglak Lee, and James Davidson. Learning latent dynamics for planning from pixels. In *ICML*, 2019.

Danijar Hafner, Timothy Lillicrap, Jimmy Ba, and Mohammad Norouzi. Dream to control: Learning behaviors by latent imagination. In *ICLR*, 2020.

Danijar Hafner, Timothy Lillicrap, Mohammad Norouzi, and Jimmy Ba. Mastering atari with discrete world models. In *ICLR*, 2021.

Nicklas Hansen, Xiaolong Wang, and Hao Su. Temporal difference learning for model predictive control. In *ICML*, 2022.

Nicklas Hansen, Hao Su, and Xiaolong Wang. TD-MPC2: Scalable, robust world models for continuous control. In *ICLR*, 2024.

Kaiming He, Haoqi Fan, Yuxin Wu, Saining Xie, and Ross Girshick. Momentum contrast for unsupervised visual representation learning. In *CVPR*, 2020.

Yann LeCun. A path towards autonomous machine intelligence. OpenReview preprint, `https://openreview.net/forum?id=BZ5a1r-kVsf`, 2022.

Bethany Lusch, J. Nathan Kutz, and Steven L. Brunton. Deep learning for universal linear embeddings of nonlinear dynamics. *Nature Communications*, 9:4950, 2018.

Lucas Maes, Quentin Le Lidec, Damien Scieur, Yann LeCun, and Randall Balestriero. LeWorld-Model: Stable end-to-end joint-embedding predictive architecture from pixels. *arXiv preprint arXiv:2603.19312*, 2026.

Seohong Park, Kevin Frans, Benjamin Eysenbach, and Sergey Levine. OGBench: Benchmarking offline goal-conditioned RL. In *International Conference on Learning Representations (ICLR)*, 2025.

Jean Ponce, Basile Terver, Martial Hebert, and Michael Arbel. Dual perspectives on non-contrastive self-supervised learning. In *International Conference on Learning Representations (ICLR)*, 2026.

Pablo Ruiz-Morales, Dries Vanoost, Davy Pissoort, and Mathias Verbeke. Koopman invariants as drivers of emergent time-series clustering in joint-embedding predictive architectures. *arXiv preprint arXiv:2511.09783*, 2025.

Yunhao Tang, Zhaohan Daniel Guo, Pierre Harvey Richemond, Bernardo Ávila Pires, Yash Chandak, Rémi 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 *International Conference on Machine Learning (ICML)*, volume 202 of *PMLR*, pp. 33632–33656, 2023.

Yuval Tassa, Yotam Doron, Alistair Muldal, Tom Erez, Yazhe Li, Diego de Las Casas, David Budden, Abbas Abdolmaleki, Josh Merel, Andrew Lefrancq, Timothy Lillicrap, and Martin Riedmiller. DeepMind Control Suite. *arXiv e-prints*, art. arXiv:1801.00690, January 2018. doi: 10.48550/arXiv.1801.00690.

Basile Terver, Tsung-Yen Yang, Jean Ponce, Adrien Bardes, and Yann LeCun. What drives success in physical planning with joint-embedding predictive world models? *arXiv preprint arXiv:2512.24497*, 2025.

Aäron van den Oord, Yazhe Li, and Oriol Vinyals. Representation learning with contrastive predictive coding. *arXiv preprint arXiv:1807.03748*, 2018.

Jeffrey Scott Vitter. Random sampling with a reservoir. *ACM Transactions on Mathematical Software*, 11(1):37–57, 1985. doi: 10.1145/3147.3165.

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

Ying Wang, Oumayma Bounou, Gaoyue Zhou, Randall Balestriero, Tim G. J. Rudner, Yann LeCun, and Mengye Ren. Temporal straightening for latent planning. *arXiv preprint arXiv:2603.12231*, 2026b.

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

Haiyu Wu, Randall Balestriero, and Morgan Levine. VISReg: Variance-invariance-sketching regularization for JEPA training. *arXiv preprint arXiv:2606.02572*, 2026.

Hao Wu and Frank Noé. Variational approach for learning Markov processes from time series data. *Journal of Nonlinear Science*, 30:23–66, 2020. arXiv:1707.04659.

Jure Zbontar, Li Jing, Ishan Misra, Yann LeCun, and Stéphane Deny. Barlow twins: Self-supervised learning via redundancy reduction. In *ICML*, 2021.

Gaoyue Zhou, Hengkai Pan, Yann LeCun, and Lerrel Pinto. DINO-WM: World models on pre-trained visual features enable zero-shot planning. In *ICML*, 2025.

A Analysis: Proofs and Discussion

A.1 Setting and notation

Throughout: linear encoder $\Phi(o) = Wo$ with $W \in \mathbb{R}^{r\times d}$, linear predictor $A \in \mathbb{R}^{r\times r}$, mean-centered observations, and sample covariances $\hat{\Sigma}_{00} = \frac{1}{T-1}\sum_t \boldsymbol{o}_t\boldsymbol{o}_t^\top$, $\hat{\Sigma}_{11} = \frac{1}{T-1}\sum_t \boldsymbol{o}_{t+1}\boldsymbol{o}_{t+1}^\top$, $\hat{\Sigma}_{01} = \frac{1}{T-1}\sum_t \boldsymbol{o}_t\boldsymbol{o}_{t+1}^\top$. Write $\Sigma_z = W\hat{\Sigma}_{00}W^\top$ and $C_z = W\hat{\Sigma}_{01}W^\top$, and assume Σ_z is invertible (equivalently, $\hat{\Sigma}_{00}$ is non-singular on the row space of W).

A.2 Proof of Theorem 1

Part 1 (without stop-gradient). The empirical loss expands as

$$\mathcal{L}(W,A) \;=\; \tfrac{1}{T-1}\sum_t \|AW\boldsymbol{o}_t - W\boldsymbol{o}_{t+1}\|^2 \;=\; \mathrm{tr}(A\Sigma_z A^\top) - 2\,\mathrm{tr}(AC_z) + \mathrm{tr}\big(W\hat{\Sigma}_{11}W^\top\big),$$

which is quadratic in A. Setting $\nabla_A\mathcal{L} = 2A\Sigma_z - 2C_z^\top = 0$ gives $A^\star = C_z^\top\Sigma_z^{-1}$ when Σ_z is invertible. Substituting,

$$\mathcal{L}^\star_{\text{no-sg}}(W) \;=\; \mathrm{tr}\big(W\hat{\Sigma}_{11}W^\top\big) \;-\; \mathrm{tr}(C_z^\top\Sigma_z^{-1}C_z).$$

Part 2 (with stop-gradient, frozen W_{tgt}). The target is $z'_{t+1} = W_{\text{tgt}}\,\boldsymbol{o}_{t+1}$, treated as constant under ∇_W. Let $\hat{\Sigma}_{zz'} = W\hat{\Sigma}_{01}W_{\text{tgt}}^\top$ and $\Sigma_{z'z'} = W_{\text{tgt}}\hat{\Sigma}_{11}W_{\text{tgt}}^\top$. The loss is

$$\mathcal{L}(W,A) \;=\; \mathrm{tr}(A\Sigma_z A^\top) \;-\; 2\,\mathrm{tr}(A\,\hat{\Sigma}_{zz'}) \;+\; \mathrm{tr}(\Sigma_{z'z'}).$$

Setting $\nabla_A\mathcal{L} = 0$ gives $A^\star = \hat{\Sigma}_{zz'}^\top\Sigma_z^{-1}$. Substituting:

$$\mathcal{L}^\star_{\text{sg}}(W;\,W_{\text{tgt}}) \;=\; \mathrm{tr}(\Sigma_{z'z'}) \;-\; \mathrm{tr}\big(\hat{\Sigma}_{zz'}^\top\Sigma_z^{-1}\hat{\Sigma}_{zz'}\big).$$

The first term does not depend on W. The second term equals $\mathrm{tr}\big(WM_{\text{tgt}}W^\top(W\hat{\Sigma}_{00}W^\top)^{-1}\big)$ where $M_{\text{tgt}} = \hat{\Sigma}_{01}\,W_{\text{tgt}}^\top W_{\text{tgt}}\,\hat{\Sigma}_{01}^\top$ is symmetric PSD and, with W_{tgt} frozen, does not depend on W. The objective is invariant under $W \mapsto RW$ for invertible $R \in \mathbb{R}^{r\times r}$, which only changes the basis of the latent space, so we optimize over r-dimensional row subspaces, equivalently over W with $W\hat{\Sigma}_{00}W^\top = I_r$. When $\hat{\Sigma}_{00}$ is singular, as when $d \geq T$, we restrict the problem to its range, where it is positive definite, since components of the rows of W in its null space vanish on every $\boldsymbol{o}_t$ entering $\hat{\Sigma}_{00}$ and $\hat{\Sigma}_{01}$ and so do not affect the fixed-target loss. Under this normalization the second term is $\mathrm{tr}(WM_{\text{tgt}}W^\top)$, a multidimensional generalized Rayleigh quotient, and by Ky Fan's variational characterization the maximizing row subspace is spanned by the top-r generalized eigenvectors of the pair $(M_{\text{tgt}}, \hat{\Sigma}_{00})$, the v with $M_{\text{tgt}}v = \mu\,\hat{\Sigma}_{00}\,v$. The optimum is a subspace rather than a unique matrix. □

A.3 Discussion of Theorem 1

Relation to CCA and VAMP. In part (1) the second term measures how much of $\boldsymbol{z}_{t+1}$'s variance is linearly predictable from $\boldsymbol{z}_t$. The squared canonical correlations between Wo_t and Wo_{t+1} are $\sum_i \rho_i^2 = \mathrm{tr}\big(C_z^\top(W\hat{\Sigma}_{00}W^\top)^{-1}C_z\,(W\hat{\Sigma}_{11}W^\top)^{-1}\big)$, which differs from the explained-variance term by the right-side whitening factor. With whitened target latents ($W\hat{\Sigma}_{11}W^\top = I_r$), the explained-variance term equals $\sum_i \rho_i^2$ and the loss is $r - \sum_i \rho_i^2$, so the objective matches CCA and the VAMP score (Wu & Noé, 2020) only up to this whitening, and sharing W across both time steps further constrains it relative to unrestricted CCA.

Symmetric versus frozen target. In part (1) the same W appears on both sides of the prediction, so the target variance $\mathrm{tr}(W\hat{\Sigma}_{11}W^\top)$ depends on W and variance the predictor cannot explain is penalized. In part (2) this term is constant, and M_{tgt} encodes the previous W, so the fixed-target optimum favors directions that predict the current latents, signal or noise. With the optimal predictor, the stop-gradient update is the gradient of $\mathcal{L}^\star_{\text{sg}}(W;W_{\text{tgt}})$ at fixed W_{tgt}, after which the algorithm sets W_{tgt} from W, and joint stop-gradient updates of encoder and predictor generally do not follow the gradient of any scalar objective (Ponce et al., 2026). If the row space of $W^\star$ spans a top generalized eigenspace of $(M_{W^\star}, \hat{\Sigma}_{00})$, with $M_{W^\star} = \hat{\Sigma}_{01}W^{\star\top}W^\star\hat{\Sigma}_{01}^\top$, then $W^\star$ is a global minimizer against its own frozen target, a self-consistency condition for full optimization against a frozen target.

Gradient dynamics can also stall elsewhere, for instance where $\hat{\Sigma}_{zz'} = 0$ makes $A^\star = 0$ and the encoder gradient vanishes. We do not characterize these stationary points or their stability. With the optimal predictor, gradient flow against the moving stop-gradient target preserves $WW^\top$ by the argument of Tang et al. (2023, Thm. 1), so stop-gradient keeps the encoder's scale, but preserved scale does not imply signal recovery (Proposition 2).

Why finite T matters. At finite T the empirical lag-1 cross-covariance $\hat{\Sigma}_{01}$ contains spurious structure, because the noise contribution $\frac{\sigma^2}{T-1}\sum_t n_t n_{t+1}^\top$ has spectral norm of order $\sigma^2\sqrt{d/T}$ for $d \ll T$, and signal–noise cross terms add further spurious structure. A one-sided objective that selects directions of the lag covariance can then align with extremal sample-noise directions when the signal's contribution is small relative to this scale. This account does not predict where the cutoff falls. For rotational dynamics with unit state covariance, the population signal lag covariance has the same norm at every frequency, so the frequency dependence in Figure 2(b,c) is an empirical finding. The closed-form predictor does not rescue stop-gradient (Figure 2d), so imperfect predictor optimization does not explain its failure.

Fixed nonlinear features. Suppose the encoder factors as $\Phi = W \circ \phi$, where the feature map $\phi : \mathbb{R}^d \to \mathbb{R}^p$ is held fixed during training, such as a frozen pretrained backbone, and only $W \in \mathbb{R}^{r\times p}$ is trained. The proof of Theorem 1 uses only the sample second moments of the input to W and the quadratic dependence of the loss on A, so both parts hold verbatim with $\hat{\Sigma}_{00}$, $\hat{\Sigma}_{11}$, and $\hat{\Sigma}_{01}$ computed over the features $\phi(\boldsymbol{o}_t)$, and the fixed-target optimum is spanned by top generalized eigenvectors in feature space. The finite-sample account transfers only qualitatively. When ϕ acts on each frame and the observation noise is independent across time, the feature residual $\phi(\boldsymbol{o}_t) - \mathbb{E}[\phi(\boldsymbol{o}_t) \mid \boldsymbol{s}_t]$ has zero mean and is independent across time given the states, so its population lag-1 cross-covariance vanishes while its sample estimate does not. This residual is neither Gaussian nor isotropic and can depend on the state, so the $\sigma^2\sqrt{d/T}$ scale does not carry over. The reduction fails when ϕ is trained or when the map on top of ϕ or the predictor is nonlinear, as in Section 5, where the connection to the theorem is empirical.

Two caveats. First, the no-stop-gradient objective is scale-degenerate: for $W_\alpha = \alpha W$ both Σ_z and C_z scale as α^2, so $\mathcal{L}^\star_{\text{no-sg}}(W_\alpha) = \alpha^2\mathcal{L}^\star_{\text{no-sg}}(W)$ and the loss vanishes as $\alpha \to 0$. The variance floor penalizes this collapse limit. Second, with $\sigma > 0$ and $d - k \geq T - 1$, the centered noise orthogonal to U almost surely has rank $T-1$, so the encoder can map the centered observations onto any centered latent trajectory using noise directions alone. For even $r < T$, centered sine and cosine trajectories at $r/2$ distinct frequencies give full-rank examples with zero linear prediction loss that satisfy either variance floor after rescaling. These objectives therefore do not guarantee signal recovery, and our evidence that RVF avoids these minima and recovers the signal is empirical.

A.4 Proof of Proposition 2

Let $\Phi^\star(o) = M_s U^\top o$ with $M_s \in \mathbb{R}^{r\times k}$ satisfying $M_s^\top M_s = I_k$, and $\tilde{\Phi}(o) = M_s V^\top o$ with $V \in \mathbb{R}^{d\times k}$ orthonormal and $V^\top U = 0$ (assuming $d \geq 2k$). We show the two claims, then compare lag correlations on training and held-out data.

Part 1: Covariance shape. With $\Sigma_o = U\,\mathrm{Cov}(s)\,U^\top + \sigma^2 I_d$ and $\mathrm{Cov}(s) = I_k$:

$$\mathrm{Cov}(\Phi^\star(o)) = M_s U^\top \Sigma_o U M_s^\top = (1+\sigma^2)\, M_s M_s^\top,$$
$$\mathrm{Cov}(\tilde{\Phi}(o)) = M_s V^\top \Sigma_o V M_s^\top = \sigma^2\, M_s M_s^\top.$$

Both are positive multiples of the same matrix, so the two encoders share their eigenvalue profile up to a global scale, their scale-invariant variance ratios, and their Pearson correlations. Rescaling the collapsed encoder by $\sqrt{1+\sigma^2}/\sigma$ makes its population covariance identical to the faithful encoder's, so covariance alone cannot identify signal recovery when the encoder gain is unrestricted.

Part 2: Ground-truth probe. Define the population coefficient of determination $R^2(z,s) = 1 - \mathbb{E}\|s - \hat{s}(z)\|^2/\mathbb{E}\|s\|^2$, where $\hat{s}(z)$ is the optimal linear predictor of s from z. For $\tilde{\Phi}$, $z = \sigma M_s V^\top n$ contains no s component, so $R^2(\tilde{\Phi}(o), s) = 0$. For $\Phi^\star$, $z = M_s s + \sigma M_s U^\top n$ and, since $M_s^\top M_s = I_k$, applying $M_s^\top$ gives the equivalent observation model $M_s^\top z = s + \sigma U^\top n$, each signal component observed with signal variance 1 and noise variance σ^2, so $R^2(\Phi^\star(o), s) = 1/(1+\sigma^2)$ per component. The gap $1/(1+\sigma^2)$ is positive at any finite noise level. On the training trajectory, finite-sample R^2 can exceed this ceiling when $d \geq T$ through memorization, which is why the population gap is measured on held-out trajectories. □

Lag correlation in and out of sample. Can temporal statistics detect collapse without state labels? With $\Gamma_s = \mathbb{E}[\boldsymbol{s}_{t+1}\boldsymbol{s}_t^\top]$, the faithful encoder has lag covariance $M_s\Gamma_s M_s^\top$ and covariance $(1+\sigma^2)\,M_s M_s^\top$, so its standardized lag-1 correlation is that of $\Gamma_s/(1+\sigma^2)$. A collapsed encoder reads noise that is independent across time, so its population lag covariance is zero. In sample, however, its directions can be selected from the training noise. With at least two latent directions, it can contain a leading sample canonical pair of $(\boldsymbol{o}_t, \boldsymbol{o}_{t+1})$ in the noise subspace, whose correlation is of order $\sqrt{(d-k)/T}$ and reaches 1 when the two lagged sample row spaces intersect. Once this correlation is comparable to $\|\Gamma_s\|_{\mathrm{op}}/(1+\sigma^2)$, an in-sample lag statistic cannot separate the encoders. The estimate is heuristic.

On a held-out trajectory with independent noise, the selected directions are fixed. The collapsed encoder's lag covariance is then zero, while the faithful encoder's remains $M_s\Gamma_s M_s^\top$, so the held-out lag correlation separates them. It must be computed on the latent vector, since per-coordinate autocorrelations see only the diagonal of the lag covariance, which can vanish when the dynamics mix coordinates. Equivalently, the optimal linear predictor's error, normalized by the latent variance, is 1 for the collapsed encoder and $1 - \|\Gamma_s\|_F^2/[k(1+\sigma^2)^2]$ for the faithful one. The held-out covariance still differs only by scale (Part 1).

Normalization and diagnostics. The unit-covariance assumption is a normalization: with $\mathrm{Cov}(s) = c\,I_k$ the constants in Part 1 become $(c+\sigma^2)$ versus σ^2 and the ceiling in Part 2 becomes $c/(c+\sigma^2)$, leaving the conclusions unchanged. Section D tracks training diagnostics alongside the held-out probe over training.

B Additional Experimental Results

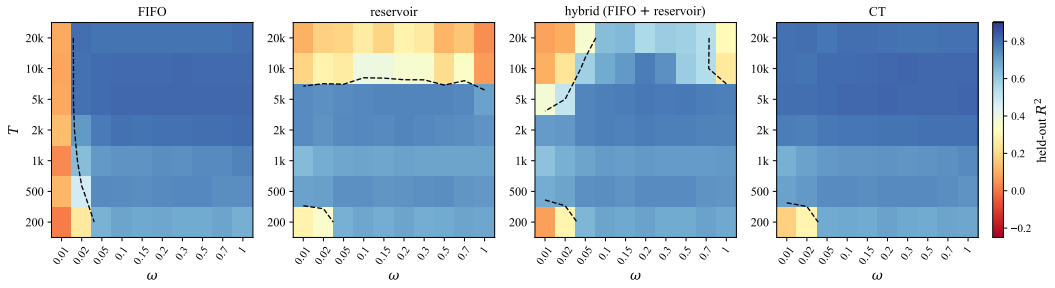


Figure 4: **Replay policies fail in complementary regions** (held-out R^2; dashed: 0.5; panels left to right). FIFO fails at its span limit; the reservoir freezes and collapses at large T; the hybrid inherits a milder collapse; CT has no failure region in the tested grid.

floor	sg	live	JEPA	ω=0.01	ω=0.05	ω=0.1	ω=0.3	ω=1
				held-out probe R^2 / ten-step rollout R^2				
$T = 500$								
oracle ceiling (R^2)				*0.83*	*0.84*	*0.84*	*0.84*	*0.84*
–	✓	✓	✓	0.10 / div.	0.14 / div.	0.04 / div.	-0.13 / div.	-0.02 / div.
slice	–	✓	✓	0.07 / div.	0.68 / 0.45	0.63 / div.	0.67 / -0.52	0.63 / div.
slice	–	–	–	**0.69** / **0.55**	0.73 / 0.67	**0.73** / **0.71**	**0.73** / 0.68	**0.73** / **0.71**
slice	–	✓	–	**0.69** / 0.50	**0.74** / **0.71**	**0.73** / **0.71**	**0.73** / **0.69**	0.72 / **0.71**
$T = 5{,}000$								
oracle ceiling (R^2)				*0.85*	*0.85*	*0.85*	*0.85*	*0.85*
–	✓	✓	✓	0.77 / -0.32	0.77 / -0.58	0.77 / -0.68	0.70 / -0.94	0.01 / div.
slice	–	✓	✓	0.05 / -0.11	**0.80** / **0.77**	0.80 / **0.79**	0.80 / 0.79	**0.81** / 0.79
slice	–	–	–	**0.79** / **0.76**	0.79 / 0.76	**0.81** / **0.79**	**0.82** / **0.80**	**0.81** / **0.80**
slice	–	✓	–	**0.79** / 0.73	**0.80** / 0.75	0.80 / 0.78	0.81 / 0.79	**0.81** / 0.79

Table 6: **Replay ablation** (circle, σ=0.3, streaming; cells as in Table 1). live=live re-encoding of stored inputs, JEPA=replay into the JEPA loss.

This appendix collects the T=5,000 twins of the main-text tables and the control axes of the streaming protocol: the i.i.d. sampling control, the data-volume axis, the noise axis, and the signal classes at T=5,000.

Protocol. Observations have dimension d=1000; the encoder is a tanh network $d \to 32 \to r$ with r=16; the streaming window of W=32 frames advances through the trajectory in time order and wraps at its end; training is SGD with learning rate 0.048 for 200,000 iterations; every cell averages 8 seeds. The sliced VF draws 128 fresh unit directions per step (results are insensitive from 8 to 512 directions). FIFO comparisons use a 128-observation buffer, and reservoir comparisons are labeled explicitly. Every regularizer's coefficient λ is swept per cell. Prediction loss and regularizer are both averaged over latent dimensions, so λ does not depend on r.

sg	floor	replay	ω=0.01	ω=0.05	ω=0.1	ω=0.3	ω=1
			held-out probe R^2 / ten-step rollout R^2				
$T = 5{,}000$							
oracle ceiling (R^2)			*0.85*	*0.85*	*0.85*	*0.85*	*0.85*
✓	–	–	0.77 / -0.44	0.77 / -0.47	0.77 / -0.59	0.71 / -0.96	0.01 / div.
✓	elem	–	0.72 / 0.66	0.72 / 0.67	0.75 / 0.71	0.76 / 0.70	0.76 / 0.61
✓	slice	–	0.73 / 0.66	0.74 / 0.67	0.76 / 0.72	0.77 / 0.73	0.76 / 0.73
✓	elem	FIFO	0.73 / 0.67	0.75 / 0.68	0.76 / 0.68	0.76 / 0.65	0.76 / 0.67
✓	slice	FIFO	0.74 / 0.69	0.76 / 0.71	0.75 / 0.72	0.77 / 0.72	0.74 / 0.71
–	elem	–	0.12 / -0.46	0.29 / div.	**0.81** / 0.60	0.78 / 0.63	0.79 / 0.76
–	slice	–	0.06 / -0.04	0.39 / -0.12	0.80 / **0.79**	**0.81** / **0.79**	**0.82** / **0.80**
–	elem	FIFO	0.18 / div.	0.75 / 0.03	0.79 / 0.53	0.79 / 0.63	0.80 / 0.72
–	slice	FIFO	0.08 / -0.09	**0.80** / **0.77**	**0.81** / **0.79**	**0.81** / **0.79**	**0.82** / **0.80**
–	slice	reservoir	0.72 / 0.71	0.74 / 0.72	0.74 / 0.72	0.75 / 0.73	0.69 / 0.69
–	elem	CT	0.73 / div.	0.72 / div.	0.78 / div.	0.80 / -0.36	0.79 / 0.69
–	slice	CT	**0.79** / **0.73**	**0.80** / 0.75	0.80 / 0.78	**0.81** / **0.79**	0.81 / 0.79

Table 7: Table 1 at T=5,000, with the reservoir row. Stop-gradient recovers slow signals with tenfold data and still fails ω=1; reservoir replay trails CT.

form	ω=0.01	ω=0.05	ω=0.1	ω=0.3	ω=1
	held-out probe R^2 / ten-step rollout R^2				
$T = 5{,}000$					
oracle ceiling (R^2)	*0.85*	*0.85*	*0.85*	*0.85*	*0.85*
SIGReg	0.77 / 0.69	0.79 / 0.56	0.80 / 0.61	0.78 / 0.71	0.79 / -0.03
VICReg	0.77 / 0.76	0.78 / 0.76	0.79 / 0.77	0.80 / 0.75	0.77 / 0.74
VP (two-sided)	0.72 / div.	0.73 / div.	0.76 / div.	0.77 / div.	0.77 / 0.18
log-det floor	**0.80** / **0.78**	**0.81** / **0.79**	**0.81** / **0.79**	**0.82** / **0.80**	**0.82** / **0.79**
RVF (elem)	0.73 / div.	0.72 / div.	0.78 / div.	0.80 / -0.36	0.79 / 0.69
RVF	0.79 / 0.73	0.80 / 0.75	0.80 / 0.78	0.81 / 0.79	0.81 / **0.79**

Table 8: Table 2 at T=5,000. The alternative forms close most of the gap and the log-det floor matches RVF; the two-sided penalty still diverges under iteration.

method	ω=0.01	ω=0.05	ω=0.1	ω=0.3	ω=1
	held-out probe R^2 / ten-step rollout R^2				
$T = 500$					
oracle ceiling (R^2)	*0.83*	*0.84*	*0.84*	*0.84*	*0.84*
baseline+sg, i.i.d.	0.19 / -0.87	0.24 / div.	0.23 / div.	-0.12 / div.	-0.02 / div.
VF (elem), i.i.d.+sg	0.49 / 0.39	0.54 / 0.53	0.53 / 0.49	0.44 / -0.08	0.39 / 0.24
VF (elem), i.i.d.	0.55 / div.	0.62 / div.	0.68 / div.	0.69 / -0.88	**0.76** / -0.39
VF (slice), i.i.d.	**0.72** / **0.69**	**0.75** / **0.73**	**0.75** / **0.75**	**0.75** / **0.73**	0.73 / **0.74**
RVF, i.i.d.	0.71 / 0.68	0.73 / 0.71	0.73 / 0.73	0.73 / 0.72	0.72 / 0.72
$T = 5{,}000$					
oracle ceiling (R^2)	*0.85*	*0.85*	*0.85*	*0.85*	*0.85*
baseline+sg, i.i.d.	0.77 / -0.21	0.77 / -0.22	0.77 / -0.37	0.70 / -0.97	0.01 / div.
VF (elem), i.i.d.+sg	0.75 / 0.67	0.75 / 0.66	0.76 / 0.72	0.76 / 0.69	0.76 / 0.71
VF (elem), i.i.d.	0.75 / div.	0.77 / div.	**0.80** / div.	0.78 / -0.50	0.80 / 0.70
VF (slice), i.i.d.	**0.81** / **0.78**	**0.80** / **0.79**	**0.80** / **0.79**	**0.81** / **0.80**	**0.81** / **0.79**
RVF, i.i.d.	0.80 / 0.76	**0.80** / 0.74	**0.80** / 0.77	0.80 / 0.78	**0.81** / 0.78

Table 9: i.i.d. minibatch control (circle, σ=0.3): the same arms under independent sampling. Replay adds nothing once minibatches are independent.

method	T=200	500	2,000	5,000	20,000
	held-out probe R^2 / ten-step rollout R^2				
ω=0.05					
oracle ceiling (R^2)	*0.82*	*0.84*	*0.85*	*0.85*	*0.85*
baseline+sg	-0.47 / -0.33	0.28 / div.	0.54 / div.	0.77 / -0.47	0.83 / 0.07
VF (elem)+sg	0.42 / 0.41	0.51 / 0.48	0.64 / **0.55**	0.72 / 0.67	**0.84** / 0.50
RVF	**0.67** / **0.66**	**0.74** / **0.71**	**0.78** / 0.38	**0.80** / **0.75**	**0.84** / **0.66**
ω=0.3					
oracle ceiling (R^2)	*0.83*	*0.84*	*0.84*	*0.85*	*0.85*
baseline+sg	-1.98 / div.	-0.13 / div.	0.50 / div.	0.71 / -0.96	0.71 / -0.96
VF (elem)+sg	0.42 / 0.40	0.53 / 0.34	0.68 / 0.60	0.76 / 0.70	0.79 / 0.65
RVF	**0.68** / **0.66**	**0.73** / **0.69**	**0.77** / **0.76**	**0.81** / **0.79**	**0.86** / **0.69**

Table 10: **Effect of trajectory length** T under streaming (circle, σ=0.3). Ceilings are finite-sample probe estimates.

sg	floor	replay	helix	lissajous	3-freq	cone	spiral
			held-out R^2, slow ω=0.1 / fast ω=1.0				
oracle ceiling (R^2)			0.98 / 0.98	0.98 / 0.98	0.98 / 0.98	1.00 / 1.00	0.71 / 0.71
✓	–	–	0.12 / 0.00	**0.97** / 0.00	**0.97** / 0.00	**0.99** / 0.01	0.05 / 0.01
✓	elem	–	0.38 / 0.84	0.95 / **0.96**	0.95 / 0.87	0.89 / **0.85**	0.23 / 0.34
–	elem	FIFO	0.61 / 0.87	0.94 / 0.95	0.94 / 0.93	0.91 / **0.85**	0.23 / 0.31
–	slice	–	0.86 / **0.91**	0.95 / **0.96**	0.94 / **0.95**	0.80 / 0.80	0.22 / 0.29
–	slice	FIFO	0.88 / **0.91**	**0.97** / **0.96**	0.95 / **0.95**	0.81 / 0.80	0.27 / 0.30
–	slice	CT	**0.97** / 0.78	**0.97** / **0.96**	0.96 / **0.95**	0.87 / 0.83	**0.40** / **0.41**

Table 11: Results on curves under streaming at T=5,000, σ=0.1.

method	ω=0.05	ω=0.3	ω=1
	held-out probe R^2 / ten-step rollout R^2		
$\sigma = 0.1$			
oracle ceiling (R^2)	*0.98*	*0.98*	*0.98*
baseline+sg	0.97 / 0.95	0.97 / 0.95	0.00 / -0.01
RVF (elem, FIFO)	0.95 / 0.65	**0.98** / 0.83	**0.98** / 0.96
RVF	**0.98** / **0.97**	**0.98** / **0.98**	**0.98** / **0.97**
$\sigma = 0.3$			
oracle ceiling (R^2)	*0.85*	*0.85*	*0.85*
baseline+sg	0.77 / -0.47	0.71 / -0.96	0.01 / div.
RVF (elem, FIFO)	0.75 / 0.03	0.79 / 0.63	0.80 / 0.72
RVF	**0.80** / **0.75**	**0.81** / **0.79**	**0.81** / **0.79**
$\sigma = 0.6$			
oracle ceiling (R^2)	*0.58*	*0.58*	*0.58*
baseline+sg	-0.00 / div.	0.00 / div.	0.00 / div.
RVF (elem, FIFO)	0.28 / div.	0.24 / -0.22	0.30 / 0.06
RVF	**0.35** / **0.20**	**0.36** / **0.30**	**0.33** / **0.32**

Table 12: **Effect of noise and frequency under streaming** (circle, T=5,000)

C EMA TARGET-ENCODER CONTROL

In practice, I-JEPA (Assran et al., 2023) and V-JEPA (Bardes et al., 2024) use an exponential moving average (EMA) target encoder rather than a plain stop-gradient copy. The EMA target W_{tgt} tracks the online encoder with momentum $\tau \in [0, 1)$: $W_{\text{tgt}} \leftarrow \tau W_{\text{tgt}} + (1 - \tau) W$. Smaller τ means a faster-tracking target; gradients to the target branch are blocked at every τ. We test whether EMA softens or eliminates the sg cutoff.

Figure 5 sweeps $\tau \in \{0.9, 0.99, 0.999\}$ on circle under the streaming protocol with the held-out probe: every EMA curve overlays the plain stop-gradient baseline, failing exactly where it fails, while RVF holds the full band.

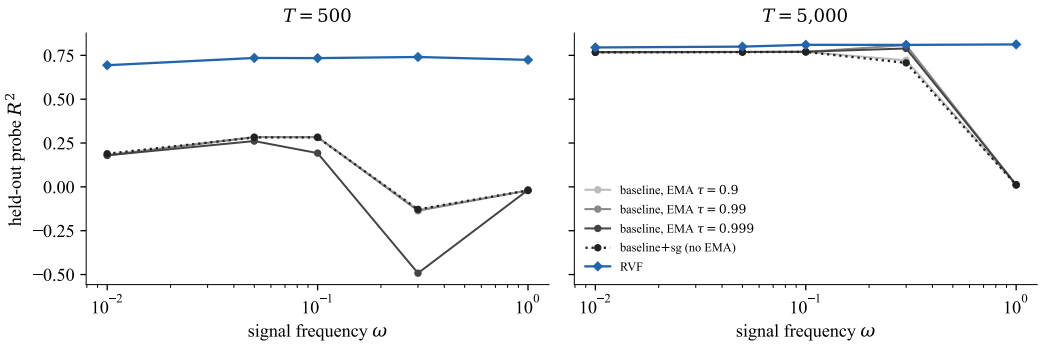


Figure 5: **EMA does not rescue stop-gradient** (circle, streaming, σ=0.3, held-out R^2, best λ per point). Grey: baseline with an EMA target at three τ values; black: plain stop-gradient; blue: RVF.

D DIAGNOSTIC BLINDNESS

Figure 6 tracks six metrics over 200,000 training iterations at ω=0.3, σ=0.3, tanh encoder, experiential W=32, 8 seeds. R^2 (top left) separates the arms: RVF recovers the signal, whereas both baseline+sg arms and the floored stop-gradient arms stay below it. The remaining five diagnostics, per-dim std, covariance shape deviation $\|\Sigma_z/\bar{\sigma}^2 - I\|_F$, mean off-diagonal correlation, trajectory curvature, and prediction loss, show no comparable separation. The encoder loses signal while no tracked training statistic flags it.

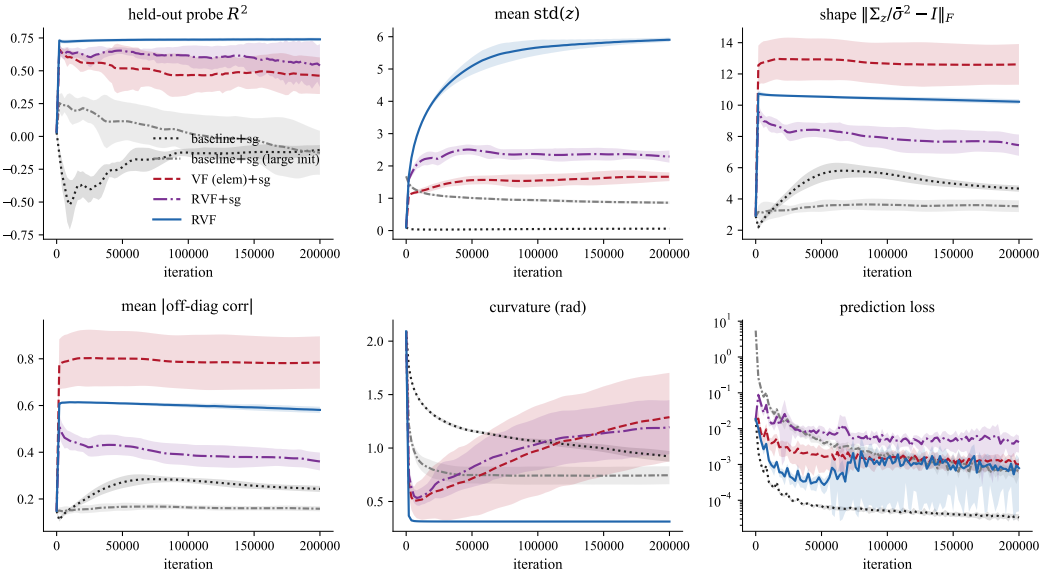


Figure 6: **Training diagnostics do not reveal signal collapse** (circle, ω=0.3, σ=0.3, streaming, 8 seeds, shaded ± 1 std). Only the held-out probe (top left) orders the methods.

E VISUAL PLANNING HYPERPARAMETERS

All visual planning evaluations use start-goal pairs from held-out trajectories and report success after 20 MPC replans, averaged over three planning seeds per training seed (Section 5.1). On both PointMaze mazes and on Reacher, a gradient-descent planner initializes the action sequence at zero and runs 100 optimization steps with learning rate 0.1 and no action noise, over a horizon of 50 environment steps, and the agent executes 5 environment steps after each replan. Goals lie 50 environment steps ahead, at least 3 maze units away on PointMaze, and each planning seed evaluates 50 start-goal pairs on PointMaze and 100 on Reacher. On Cube, a cross-entropy-method planner samples 300 action sequences, refits to the top 30 for 10 iterations over a horizon of 25 environment steps, and executes 5 environment steps after each replan, over 100 episode-balanced start-goal pairs.