SEMIGROUP-JEPA: LATENT DYNAMICS CONSISTENCY FOR ZERO-SHOT PHYSICS GENERALIZATION

Anonymous authors

Paper under double-blind review

ABSTRACT

Joint-Embedding Predictive Architecture (JEPA) world models learn a compact latent representation of the world that supports prediction and planning, but their capability to learn physics and generate physically realistic dynamics remains hitherto untested. In this work, we introduce *SemiGroup-JEPA (SG-JEPA)*, which extends the LeWorldModel framework by supplying the parameter governing the physics to the temporal model via action-conditioning and jointly training an encoder and predictor through an autoregressive latent rollout. To evaluate the model's ability to generalize out of distribution, we design dynamical tasks under different gravitational fields that, despite obeying the same physical law, exhibit qualitatively different dynamics, ranging from floating motion in weak gravitational fields to rapid bouncing in strong ones. In contrast to DINO-WM, SG-JEPA reduces open-loop prediction error by up to $2\times$ on two-dimensional datasets, and increases control success rate up to $2.5\times$ for three-dimensional robotic datasets, for which we train independent diffusion policies. To explain this advantage, we develop a linear feature model that separates local law-conditioned error from its recursive amplification under rollout. Guided by this model, we find that back-propagating the multi-step rollout loss into the representation trains the encoder to keep the features that the predictor can carry forward, and that those are the features the dynamics depend on, so most of the gain comes from the encoder learning better features rather than from the predictor learning better dynamics.

1 INTRODUCTION

The long-standing goal of world models is to understand dynamics of the environment in which the model agent is deployed so that the agent can predict subsequent states and consequences of its actions. Joint-Embedding Predictive Architecture (JEPA) emerged as a promising approach, whereby rather than reconstructing future pixels, it learns latent representations by jointly training an encoder and predictor (LeCun, 2022; Sobal et al., 2025; Zhou et al., 2025). Recent proposals such as DINO-WM (Zhou et al., 2025) and LeWorldModel (LeWM) (Maes et al., 2026) have demonstrated that these latent predictions support downstream planning and control under a fixed frame-to-frame transition law. We push this further by asking the question:

Has the model actually learned the physical dynamics when its learned update is composed over long horizons? Does it generalize to out-of-distribution (OOD) dynamics?

If the answer is yes, the consequences are significant. A world model that genuinely learns dynamics would enable an agent to adapt in new environments, e.g. a model trained on Earth dynamics generalizing to Martian dynamics.

In this work, we take gravitational dynamics as the primary testbed for these questions, and introduce *Semigroup-JEPA (SG-JEPA)*, a gravity-conditioned extension of LeWM with recursive physical prediction (Fig. 1). SG-JEPA supplies gravity alongside action and jointly trains its encoder and predictor through a discounted autoregressive rollout loss, while SIGReg (Balestriero & LeCun, 2025) regularizes the latent space. On various MuJoCo datasets covering two-dimensional (2D) rigid body freefall, three-dimensional (3D) projectile motion and robot-arm control, we train SG-JEPA on a

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.

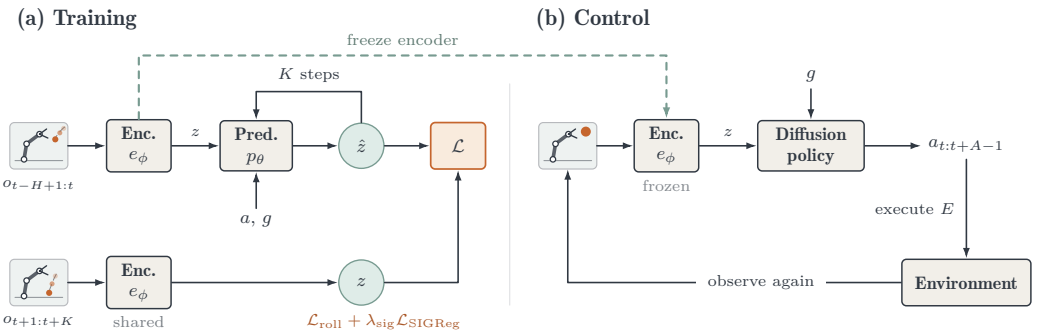


Figure 1: **SG-JEPA training and control.** (a) The encoder and gravity-conditioned predictor are jointly trained with a discounted K-step autoregressive rollout loss and SIGReg. Targets use the same trainable encoder. (b) A gravity-conditioned diffusion policy is trained on demonstrations from successful episodes with the encoder frozen. At inference, it generates $a_{t:t+A-1}$, executes the first E actions, and replans from new observations.

narrow range of gravity values and evaluate on a wider grid that extends far beyond it. We study two aspects of generalization, namely transfer to OOD gravity values and accuracy under long-horizon rollout. Our results show that the dynamics SG-JEPA learns within a narrow gravity range remain accurate over long horizons at OOD gravity values, demonstrating zero-shot physics generalization.

Across these gravitational environments we find that how well the encoder preserves dynamical information determines the ability to generalize out of distribution. To interpret this behavior, we mechanistically isolate the encoder from the predictor: starting from a trained checkpoint, we freeze each encoder and fit freshly initialized predictors to it. The advantage follows the encoder trained jointly with the GRU predictor, whichever fresh predictor is fitted, and it widens as gravity moves further from the training range and as the rollout horizon grows.

Related work. JEPA models introduce a new paradigm of learning by predicting target representations rather than reconstructing pixels, and have been extended from images to video (LeCun, 2022; Assran et al., 2023; Bardes et al., 2024; Assran et al., 2025). DINO-WM (Zhou et al., 2025) applies this idea with frozen DINOv2 features (Oquab et al., 2024), while LeWM (Maes et al., 2026) jointly trains its encoder and predictor with SIGReg (Balestriero & LeCun, 2025). Recent work probes physics generalization under held-out configurations and asks which physical parameters predictive latents retain (Xue et al., 2026; Tan et al., 2026). Our setting differs in isolating gravity as a physical quantity governing the dynamics, which lets us systematically test the learned dynamics outside the training range. A separate line of work addresses rollout error by training on the model's own predictions or optimizing beyond one step (Bengio et al., 2015; Venkatraman et al., 2015; Lambert et al., 2021). In this work, we address these two problems jointly and introduce SG-JEPA, which back-propagates a recursive latent rollout loss through both the encoder and predictor, enabling, as we show below, OOD generalization over long-horizon dynamics. The name reflects that, for action-free trajectories at a fixed gravity value, the repeated updates form a discrete semigroup. Related operator-based analyses of how local representation and predictor errors propagate appear in (Williams et al., 2015; Lusch et al., 2018). Appendix A provides more discussion.

2 Problem Setting

In this section, we introduce Semigroup-JEPA (SG-JEPA). We first describe the architecture and training loss. We then introduce the customized, gravity-controlled MuJoCo environments and the baseline models we compare against. Finally we explain evaluation protocols for prediction of physical quantities and latent control.

2.1 MODEL ARCHITECTURE

Semigroup-JEPA. In Fig. 1 we sketch the architecture of SG-JEPA. It keeps the two components of LeWM, an encoder and a predictor. At time step t, the model has observation o_t and control u_t. The observation o_t is then encoded using a Vision Transformer (Dosovitskiy et al., 2021) into low-dimensional latents z_t. Unless stated otherwise, we use the ViT-Tiny configuration, with 12 Transformer layers, 3 attention heads, and hidden width 192. We use 8×8 patches for 128×128 resolution and 16×16 for 256×256 resolution. The encoder is trained from random initialization jointly with the predictor. We use the final layer `[CLS]` token $h_t \in \mathbb{R}^{192}$ as a global summary of the observed frame o_t and map it to $z_t \in \mathbb{R}^{256}$ using a one-hidden-layer projector. The latents z_t are then fed into the predictor. Gravity enters the model through the action. Each episode has a signed gravity scalar parameter g, which we treat as an extra coordinate of the action, and maintain constant across frames, so that $\tilde{a}_t = [u_t; g]$ and $c_t = q_\psi(\tilde{a}_t)$, where q_ψ is the action encoder, a temporal convolution followed by a SiLU MLP, which yields $c_t \in \mathbb{R}^{256}$. The input g is z-scored using statistics from the training split. In the case where there are no external actions, e.g. free-fall, we still input a single scalar g in the action conditioning.

We consider three choices of predictor, namely a Transformer, a GRU, and a state-space model (SSM). Unless otherwise noted, SG-JEPA refers to either the GRU or the SSM variant. The Transformer predictor has 6 layers and 16 attention heads, in line with the Original LeWM, with actions fed into every block through Adaptive Layer Normalization (AdaLN) (Peebles & Xie, 2023). The GRU predictor defaults to 3 residual layers of width 512. In each layer, the hidden state is concatenated with the projected action embedding, passed through an MLP, and then processed by a single-layer GRU. The SSM predictor uses the same depth, width, and action-concatenation scheme, replacing each GRU with a Mamba/S6-style selective state-space block (Gu & Dao, 2024). The predictor takes as input a history of length H and predicts the next latent, $\hat{z}_{t+1} = p_\theta(z_{t-H+1:t}, c_{t-H+1:t})$. Exact architectures and gravity-fusion ablations are given in Appendix C.

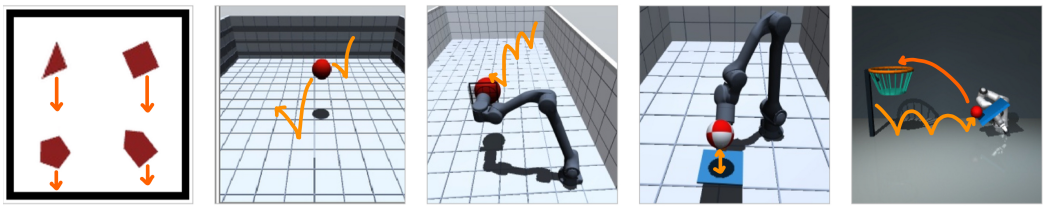
Training objective. For fixed g and an action-free trajectory, iterating the shared history update $F_{\theta,g}$ gives $S_{\theta,g}(k) = F_{\theta,g}^{\circ k}$, with $S_{\theta,g}(0) = I$ and $S_{\theta,g}(k+\ell) = S_{\theta,g}(\ell) \circ S_{\theta,g}(k)$. These iterates form a discrete semigroup on the latent history. Without controls, consecutive update blocks compose into this semigroup. With controls, they still compose, but each block is modified by its action (Appendix H.6).

SG-JEPA combines a K-step autoregressive rollout loss with SIGReg. Unlike LeWM's one-step teacher-forced objective, in which the predictor is always given true encoded latents as input rather than its own predictions, it starts from H encoded frames and recursively predicts K future latents, inserting each prediction into the next length-H history window. With $z_s^{\text{roll}} = z_s$ for $s \le t$ and $z_s^{\text{roll}} = \hat{z}_s$ for $s > t$, the rollout is $\hat{z}_{t+k} = p_\theta\big(z^{\text{roll}}_{t+k-H:t+k-1}, c_{t+k-H:t+k-1}\big)$, with $k = 1, \dots, K$. We use the normalized rollout loss

$$\mathcal{L}_{\text{roll}} = \sum_{k=1}^{K} w_k \left\| \hat{z}_{t+k} - z_{t+k} \right\|_2^2, \quad \text{where} \quad w_k = \frac{\gamma^{k-1}}{\sum_{j=1}^{K} \gamma^{j-1}}.$$

By construction, a composition law emerges from repeatedly applying the shared predictor. For $\gamma < 1$, later predictions receive less weight, so accumulated errors do not dominate the objective. The full regularized objective is $\mathcal{L} = \mathcal{L}_{\text{roll}} + \lambda_{\text{sig}} \mathcal{L}_{\text{SIGReg}}$. SIGReg acts only on encoded latents and encourages their random one-dimensional projections to match a standard Gaussian. Target latents come from the same trainable encoder, without stop-gradient.

Training setup. For training, we use a hybrid Muon/AdamW optimizer. Muon (Jordan et al., 2024) applies an orthogonalized update to matrix-valued parameters, so all 2D tensors are optimized with Muon at a learning rate of 1×10^{-4} and all other tensors with AdamW at 5×10^{-5}. In Appendix D.1.2 we perform a controlled comparison against pure AdamW (Loshchilov & Hutter, 2019), finding that Muon lowers the validation loss more rapidly and uses input data more efficiently. We train all models for 20 epochs and, unless stated otherwise, use $H = 20$, $K = 5$, and $\gamma = 0.95$; the sweeps in Appendices D.2.1, D.3, and D.1.3 justify the history length, the rollout discount, and the SIGReg coefficient, and Appendix C gives the complete training configuration.



(a) 2D shapes (b) Approach Ball (c) Arm Catcher Ball (d) Arm Paddle Ball (e) Franka Paddle-to-Basket

Figure 2: **Evaluation environments.** (a) Planar shapes: each object is given an impulse at $t = 0$ and then fall and collide freely after inside the box. (b) Approach Ball: projectile flight from the far end of the scene. (c) Arm Catcher Ball: a robotic arm catches a ball with a catcher. (d) Arm Paddle Ball: the same arm sustaining bounces with a tilting paddle. (e) Franka Paddle-to-Basket: a Panda arm strikes an incoming ball into a basket. Orange arrows sketch the motion.

2.2 Datasets

We generate eight datasets in MuJoCo (Todorov et al., 2012), covering passive rigid-body motion (freefall), projectile motion, and robot-arm control. We use six of them as environments for the experiments in the main text and the other two (2D pentagon and house) in the ablation studies in Appendix E.2. Fig. 2 gives an overview and illustrates the ball dynamics. For training, g is sampled from a narrow Gaussian, $\mathcal{N}(4, 0.5^2)$ for the 2D planar shapes and Arm Catcher Ball, and $\mathcal{N}(9.8, 2.0^2)$ for the remaining 3D datasets. Held-out test sets use a wider grid of g values, including OOD ones. Five datasets test prediction: four planar shapes receive one initial impulse and then move freely inside a box, while Approach Ball contains 3D projectile motion. Three datasets test control: Arm Catcher Ball, Arm Paddle Ball, and Franka Paddle-to-Basket. Appendix B gives the exact data splits, action schemas and other dataset details.

Baselines. We compare the performance of SG-JEPA against the following baselines. We first compare against the *Original LeWM*, the vanilla model of (Maes et al., 2026), which uses a one-step latent prediction loss, SIGReg on the encoded latents, and a causal Transformer predictor. We also compare against *DINO-WM* (Zhou et al., 2025), for which we keep the pretrained DINOv2 encoder frozen and train an action-conditioned Transformer predictor, with action conditioning performed in the same way as in LeWM. We label the GRU- and SSM-predictor variants of SG-JEPA *SG-JEPA (GRU)* and *SG-JEPA (SSM)*, respectively.

2.3 Downstream evaluation

Prediction. For prediction tasks, we freeze the trained world model and fit an MLP probe r_η that, given a short latent window, predicts the physical state s_t (position, velocity, and rotation in 2D; position and velocity in 3D) at time t. The probe is trained only on training episodes (one episode is a video sample). On held-out episodes we apply it to both encoded ground-truth frames and open-loop autoregressive rollouts at forecast horizons $\mathfrak{h}$ up to $\mathfrak{h}_{\max} = 44$ frames. Let ℓ_r denote the probe-window length. We report the excess physical-state error

$$\frac{1}{\mathfrak{h}} \sum_{k=1}^{\mathfrak{h}} \left(\mathrm{NMSE}\big(r_\eta(z^{\mathrm{roll}}_{t+k-\ell_r+1:t+k}), s_{t+k}\big) - \mathrm{NMSE}\big(r_\eta(z_{t+k-\ell_r+1:t+k}), s_{t+k}\big) \right), \tag{1}$$

where NMSE denotes the normalized mean-squared error, with state coordinates normalized using statistics from the probe's training set. Subtracting the probe error on encoded ground-truth latents isolates the error introduced by the rollout.

Control. For each control task and source model, we freeze the visual encoder and train a separate gravity-conditioned Diffusion Policy (Chi et al., 2023) on its features. SG-JEPA and LeWM provide projected CLS latents, while DINO-WM provides mean-pooled frozen DINOv2 patch features. Fig. 1(b) shows the resulting control loop. A two-layer causal GRU compresses the most recent H feature vectors into $\bar{h}_t = \mathrm{LayerNorm}(\mathrm{GRU}(z_{t-H+1:t})_{\mathrm{final}})$, where LayerNorm denotes layer

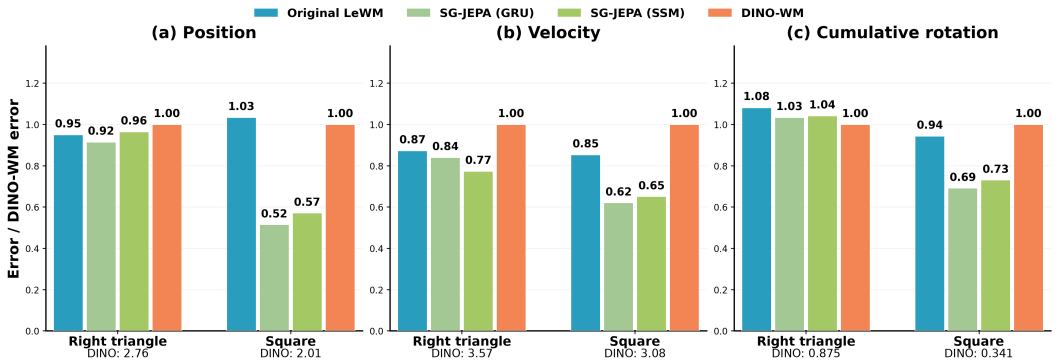


Figure 3: Long-horizon prediction at 44 rollout steps for 2D datasets, evaluated with MLP probes. Bars show position and velocity L_2 error and cumulative-rotation mean-absolute error (MAE), normalized by the DINO-WM error for each shape and metric. The absolute DINO-WM errors are shown below each group. Results are averaged over held-out episodes for each shape across 25 gravity values from -2 to 10, with training gravity sampled from $\mathcal{N}(4, 0.5^2)$. Lower is better.

normalization, and the policy is conditioned on $\xi = [\bar{h}_t; g_{\text{pol}}]$, with g_{pol} the z-scored gravity. The noise predictor ϵ_ω is a conditional 1D U-Net, trained on action blocks $\mathbf{u} = u_{t:t+A-1}$ with the standard denoising objective of Chi et al. (2023); Appendix C.2 states it. We train the GRU and U-Net while keeping the visual encoder frozen. All policies use $H = 20$, $A = 16$, a 256-dimensional GRU, and a 100-step cosine diffusion schedule. At inference, the policy predicts A controls, executes the first E, and replans from the new observation. We use $E = 8$ for Arm Catcher Ball and Franka Paddle-to-Basket and $E = 4$ for Arm Paddle Ball. Note that this execution interval is distinct from the world-model training rollout length K.

Our tasks differ from static-goal benchmarks such as Two Rooms and Push-T in that the ball moves, so the policy must infer its future motion from the observation history and gravity. A goal-conditioned Cross-Entropy Method planner would need a future target frame, which would give away the ball's future position, so our policies do not use one.

3 Empirical Results

In this section, we compare SG-JEPA against LeWM and DINO-WM across our 2D and 3D datasets. SG-JEPA improves long-horizon prediction and achieves higher average control success rates than the Original LeWM and DINO-WM, particularly at OOD gravity values.

2D datasets. In Fig. 3 we present long-horizon physical-state prediction results. Each episode contains 64 frames and our models take a history of $H = 20$, so at test time we roll out for 44 steps and probe the position, velocity, and rotation. Note that rotation is computed by integrating the probed angular velocity over the rollout, and its error is measured in number of turns. Complete per-gravity and error-by-horizon results are detailed in Appendix E. On the square object, we find that both variants of SG-JEPA have a clear advantage: relative to DINO-WM, SG-JEPA (GRU) reduces the three error types by 31–48%, with SG-JEPA (SSM) close behind, while Original LeWM remains near the DINO-WM baseline. The advantage is most evident over longer rollouts (Appendix Table 8). An SG-JEPA variant leads every square metric from 5 through 44 rollout-steps. On triangles, although DINO-WM wins by a small margin for short-horizon position and velocity, SG-JEPA achieves the lowest 44-rollout-step errors on both. This behavior is consistent with rollout training reducing the accumulation of errors when predictions are fed back into the model. The triangle dataset remains a challenging case: DINO-WM has slightly lower cumulative-rotation error, which we attribute mostly to asymmetric contacts amplifying small errors after collisions. As for OOD generalization, SG-JEPA has the lowest error across most of the held-out gravity grid on the square dataset, with significant advantages in velocity and rotation for large gravity parameter values (Appendix Fig. 18). Beyond gravity, we also test generalization to an unseen shape. We train SG-JEPA on triangles and squares and evaluate frozen-model rollouts on a "house" dataset, formed by attaching the right triangle to

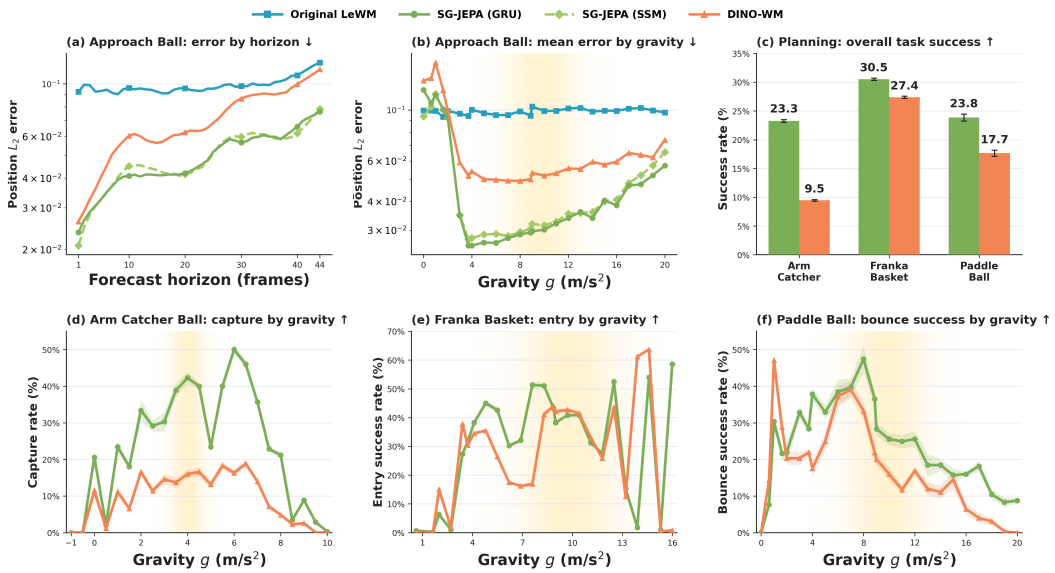


Figure 4: 3D prediction and control tasks across held-out gravity values. *Open loop*(a)–(b): Approach Ball position L_2 error on held-out episodes with a trained probe. *Closed loop*(c)–(f): Evaluation results for Diffusion Policies trained on frozen world-model features, aggregated by task and resolved by gravity (higher is better). Results are averaged over five seeds. Arm Catcher Ball and Franka Paddle-to-Basket use $A = 16$, $E = 8$; Arm Paddle Ball uses $A = 16$, $E = 4$. Yellow shading marks the training-gravity distribution, $\mathcal{N}(4, 0.5^2)$ for Arm Catcher Ball and $\mathcal{N}(9.8, 2.0^2)$ for the other two.

the square along the triangle's longest side, using a probe trained on house states. The frozen model still yields a useful representation of the house state, and its 44-rollout-step position and velocity predictions transfer reasonably well, particularly with a larger ViT encoder. A detailed breakdown is given in Appendix E.2.

3D datasets. *Prediction.* On the Approach Ball dataset, we find that the GRU and SSM SG-JEPA variants reduce mean position error by approximately 34% relative to DINO-WM and 50% relative to Original LeWM (Appendix Table 10). From Fig. 4(a), DINO-WM performs comparably well at short horizons but drifts more quickly during rollout. Both SG-JEPA variants remain more accurate over longer rollouts. The GRU and SSM variants perform similarly and alternate as the best method across horizons, suggesting that the improvement does not depend on a single temporal architecture. From Fig. 4(b) we see that this pattern holds across gravity values. SG-JEPA is best at 22 of the 25 test values, with the exceptions being for very small gravity values, where the Original LeWM performs best. In Appendix F.1 we present more results and observe similarly that SG-JEPA retains its advantage for velocity generalization.

Control. For control tasks we evaluate each frozen encoder by training a task-specific Diffusion Policy on its latents, run in the receding-horizon loop of Sec. 2.3, so control is open-loop within each action block but closed-loop over the episode. Success is defined by task-specific physical events: Arm Catcher Ball requires the ball to enter the central area of the catcher and be latched before the episode ends; Franka Paddle-to-Basket requires a valid paddle hit followed by the ball landing in the basket; Arm Paddle Ball requires that the ball never hit the floor and that at least one bounce reach a $0.28\,\mathrm{m}$ apex, which rules out the degenerate case of the ball resting on the paddle.

In Fig. 4(c) we show the aggregate results for the three tasks and (d)–(f) show their per-gravity breakdown. Each episode is evaluated with 5 policy rollouts, and we plot the mean and standard deviation. To keep the main figure readable, we present only the SG-JEPA (GRU) vs. DINO-WM comparison, as DINO-WM provides a stronger baseline for these tasks. Full results including Original LeWM can be found in the relevant appendices. Below we summarize our main observations.

- The largest and most consistent gain occurs for Arm Catcher Ball, where SG-JEPA (GRU) raises capture success from 9.5% to 23.3%. The gain peaks around the training-set values and extends

from $g = 0$ to $g = 9$. A success peak is observed at $g = 0$ likely because for this gravitational field the ball is not bouncing so the trajectory is relatively easy to predict. On the other hand, for strong gravitational fields the ball bounces frequently, making it difficult to accurately predict the dynamics.

- The Franka Paddle-to-Basket results are less uniform across gravity values. The advantage of SG-JEPA (GRU) appears after impact. DINO-WM achieves a slightly higher paddle-hit success rate (both achieve above 95% hit rates; Appendix Fig. 22) but SG-JEPA (GRU) converts more contacts into basket entries, raising average success from 27.4% to 30.5%. Thus the main difficulty lies in producing the post-contact position and velocity that send the ball into the basket. At low g the struck ball can rise for too long and miss the basket, whereas at high g a stronger and more precise impulse is needed to reach it.
- For Arm Paddle Ball, SG-JEPA (GRU) raises success from 17.7% to 23.8%. SG-JEPA (GRU) both avoids floor contact more often and converts more retained episodes into qualifying bounces. At $g = 0$, both policies enable the arm to retain the ball but not to complete a bounce because for this value of g an upward-moving ball never falls back down. For high g, the flight time $2v_z/g$ and apex height $v_z^2/(2g)$ both shrink, leaving less time to reposition the paddle and making the $0.28\,\mathrm{m}$ apex threshold harder to reach.

Overall, gravity exposes different bottlenecks in each task: interception timing, post-impact targeting, or repeated contact regulation. Together, they strengthen our claim that SG-JEPA's representations better preserve the dynamics and generalize better OOD than existing baselines. Complete 3D dataset results are in Appendices F.2, F.3, and F.4.

Takeaway. We propose *Semigroup-JEPA* (SG-JEPA), a gravity-conditioned latent world model that jointly trains the encoder and predictor over recursive latent rollouts, demonstrating performance gains over Original LeWM and DINO-WM. Relative to DINO-WM, for example, it reduces position prediction error by 34% on *Approach Ball* and improves closed-loop capture success from 9.5% to 23.3% on *Arm Catcher Ball* across a range of in-distribution and OOD gravity values.

4 Where Does the Long-Horizon Advantage Come From?

SG-JEPA's long-horizon OOD advantage can arise at two stages. A difference may already be present in the one-step prediction (which we refer to as the local transition in the theoretical model below) at an unseen gravity value, before any prediction is fed back, and recursive rollout can then amplify or shrink that difference. In this section we use a linear feature model to separate these stages, which tells us what to measure to locate the advantage.

Theoretical model. Let $\phi_t \in \mathbb{R}^{d_\phi}$ be a feature state with dynamics $\phi_{t+1} = T(g)\phi_t + \xi_{t+1}$, where $T(g)$ is the one-step transition at gravitational parameter g and $\mathbb{E}[\xi_{t+1} \mid \phi_t, g] = 0$. Let $z_t = W\phi_t \in \mathbb{R}^{d_z}$ be the learned latent. We assume $d_z \le d_\phi$ and $W \in \mathbb{R}^{d_z \times d_\phi}$ is row-orthonormal, with $WW^\top = I_{d_z}$, and define $A_W(g) = WT(g)W^\top$ and $C_W(g) = WT(g) - A_W(g)W$. When a teacher-forced predictor $\widehat{A}(g)$ starts from the true latent, its conditional-mean defect is

$$\delta_t(g) = C_W(g)\phi_t + [A_W(g) - \widehat{A}(g)]z_t. \tag{2}$$

The closure term $C_W(g)\phi_t$ is the part of the next latent that still depends on features discarded by W. If $C_W(g) = 0$, the representation is predictively closed at gravity g: the current latent contains all information needed for the conditional mean of the next latent. The realized teacher-forced error is $z_{t+1} - \widehat{z}^{\mathrm{TF}}_{t+1} = \delta_t(g) + W\xi_{t+1}$, where $\widehat{z}^{\mathrm{TF}}_{t+1} = \widehat{A}(g)z_t$. Composition at a fixed gravity and transfer across gravity values are separate questions. Suppose $T(g) = \sum_{k=1}^{m} \psi_k(g)T_k$ and the projected and learned operators share this finite law basis. Let $\psi(g) = (\psi_1(g), \dots, \psi_m(g))^\top$, $M_\psi = \mathbb{E}_{g\sim P_{\mathrm{tr}}}[\psi(g)\psi(g)^\top] \succ 0$, and $\mathcal{L}_{\mathrm{law}}(g_\star) = \psi(g_\star)^\top M_\psi^{-1}\psi(g_\star)$, which measures how well the test gravity value $g_\star$ is covered by the training distribution. Appendix H.3 proves that

$$\|\delta_t(g_\star)\|_2 \le \sqrt{\mathcal{L}_{\mathrm{law}}(g_\star)}\,(B_z\epsilon_{\mathrm{op}} + B_\phi\epsilon_{\mathrm{cl}}), \tag{3}$$

where B_z and B_ϕ bound the latent and feature states, while ϵ_{op} and ϵ_{cl} are the training-average predictor and closure errors. For free flight, gravity enters the transition affinely, so $\psi(g) = (1, g)^\top$

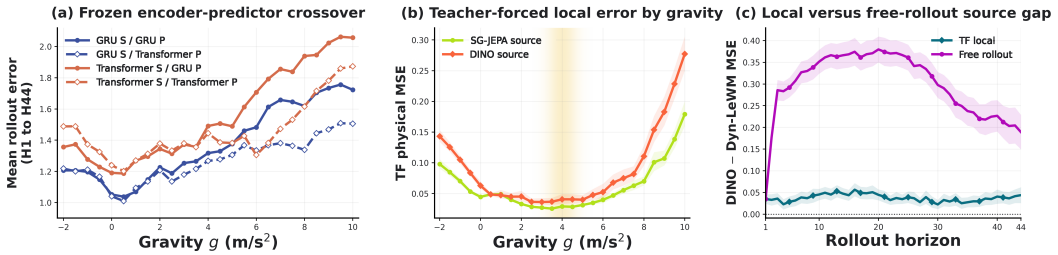


Figure 5: (a) Frozen-source predictor crossover on held-out 2D square episodes. Color denotes source; line style denotes fresh predictor. (b) Teacher-forced physical MSE from true $H = 20$ context, with three predictor seeds and pointwise 95% bootstrap intervals. (c) DINO-minus-Dyn teacher-forced and free-rollout gaps over 2,800 paired OOD episodes, with 95% simultaneous intervals. Positive values favor SG-JEPA.

and $\mathcal{L}_{\text{law}}(g_\star) = 1 + \frac{(g_\star - \mu_{\text{tr}})^2}{\sigma_{\text{tr}}^2}$. The bound therefore grows with the distance of $g_\star$ from the training gravity mean. Its main implication is that gravity conditioning alone does not guarantee transfer: unseen-gravity accuracy also depends on law coverage, predictor fit, and representation closure.

From local to long-horizon error. At fixed g, $S_g(\mathfrak{h}) = T(g)^{\mathfrak{h}}$ and $\widehat{S}_g(\mathfrak{h}) = \widehat{A}(g)^{\mathfrak{h}}$ form the true and learned discrete evolution semigroups. Starting from the same latent, the free-rollout error $e_{\mathfrak{h}} = z_{\mathfrak{h}} - \widehat{z}_{\mathfrak{h}}$ satisfies

$$e_{\mathfrak{h}} = \sum_{j=0}^{\mathfrak{h}-1} \widehat{A}(g)^{\mathfrak{h}-1-j} \left[\delta_j(g) + W\xi_{j+1}\right]. \tag{4}$$

An error introduced at step j is therefore propagated through the $\mathfrak{h} - 1 - j$ learned updates that follow it. Importantly, one-step errors of the same size can have very different consequences after rollout, depending on how the learned updates transform them. A one-step objective is blind to this difference, whereas a multi-step objective penalizes it directly. Appendix H.2 formalizes this through a semigroup-intertwining defect theorem, and Appendix H.7 treats dynamical regime changes at events such as collisions.

The theoretical model suggests two empirical tests. First, if SG-JEPA's advantage lies in the encoder's representation, it should survive replacing the original predictor with a freshly trained one, and it should already be visible under teacher forcing, before any predicted latent is fed back. Second, if feeding predictions back matters, the performance gap between encoders should differ between teacher forcing and free rollout, although the sign of that difference and how it varies with horizon are empirical questions.

Empirical observations. Fig. 5 applies these two tests to the 2D square dataset. Panel (a) freezes the encoders learned jointly with the GRU and Transformer predictors, then fits a new GRU and a new Transformer predictor to each encoder. Panel (b) freezes the SG-JEPA (GRU) and DINO-WM encoders, fits a fresh GRU predictor of matched size to each, and measures teacher-forced mean-squared error (MSE) on collision-free transitions. Panel (c) compares this teacher-forced gap with the gap obtained when predictions are fed back recursively. The main takeaways are as follows.

- **The advantage lies in the GRU-trained representation.** Panel 5(a) discards the original predictors and fits the same fresh GRU and Transformer architectures to each frozen encoder. With either predictor, the GRU-trained encoder lowers mean rollout error by about 12% relative to the Transformer-trained encoder: 1.376 versus 1.555 with a fresh GRU, and 1.269 versus 1.453 with a fresh Transformer. That the ranking of the two encoders does not depend on which fresh predictor is fitted shows that the gain follows the representation learned during GRU training, rather than the original GRU predictor.
- **SG-JEPA is better before recursive feedback.** Panel 5(b) tests each local transition before predicted latents are reused. For each frozen encoder, we fit a fresh GRU and restore the true $H = 20$ context before every collision-free prediction. SG-JEPA has the lower local-error point estimate at all 25 test gravity values. On the far-OOD gravity values $g \leq 2$ or $g \geq 6$, it reduces local error by about 32% relative to DINO-WM. Applying the same physical-state probe to the

true and predicted next latents removes the probe error common to both models. The advantage therefore arises before rollout error accumulates.

- **Recursive rollout widens the advantage.** In Panel 5(c), positive values favor SG-JEPA. From horizon 2 onward, the free-rollout gap is several times larger than the teacher-forced gap, peaking near 0.38 around horizon 20, while the local gap remains below 0.06. This is consistent with Eq. 4, which says that once predictions are fed back, a small one-step difference between the encoders grows into a much larger gap. That gap shrinks again at long horizons, so the amplification is not monotone.

Ablations. Ablations are presented in the appendices. Appendix D.4 tests whether the predictor actually uses gravity through an evaluation-time counterfactual, in which we hold the physical trajectories fixed and feed the predictor a gravity value that differs from the true one. For both GRU and Transformer predictors, the correct gravity minimizes rollout error, while larger mismatches generally increase it. Appendix G studies adaptation from sparse post-training data. We post-train on disjoint support episodes at $\mathcal{G}_{\text{sup}} = \{0, 2, 6, 8\}$ and evaluate on 13 unseen interpolation gravity values. We find that post-training reduces SG-JEPA (GRU) error by 13.6% on the square and 20.8% on the triangle. Across all eight model-shape settings, it improves on the pretrained model and outperforms post-training on the target gravity alone, averaging a 14.1% reduction versus 6.9%.

Takeaway. SG-JEPA's OOD advantage lies in the representation the encoder learns during GRU training. We know this because the advantage survives replacing the predictor and is already present before any feedback, where it lowers far-OOD one-step error by 32%. The linear feature model explains why such a small one-step edge matters. Gravity conditioning alone does not guarantee transfer, and once predictions are fed back, a small one-step difference between encoders grows into a much larger long-horizon gap, which is what we see under recursive rollout.

5 Conclusion

In this work, we introduce Semigroup-JEPA (SG-JEPA), which combines gravity conditioning with joint encoder–predictor training through recursive latent rollouts. To test whether a world model learns physical dynamics rather than a single transition law, we build eight MuJoCo datasets in which gravity is sampled from a narrow band at training time and from a much wider grid at test time. On 2D and 3D prediction and robot control, SG-JEPA outperforms Original LeWM and DINO-WM, with the largest gains at long horizons and far from the training gravity, and the same frozen representation raises Diffusion Policy success rates on all three manipulation tasks. To explain these gains, we develop a linear feature model that separates local law-conditioned error from its amplification under rollout and bounds the one-step error at an unseen gravity value by how well the training gravities cover it. Guided by this model, we train fresh predictors on frozen checkpoints and find that SG-JEPA's advantage lies in the encoder's representation, which recursive rollout then amplifies over the horizon.

Limitations. Several important directions remain for future work. First, our experiments vary a single scalar, gravity. It would be of interest to study a setting with vector-valued physical variables, or even to have the model infer dynamical parameters directly from observation. Second, transfer across object shapes is uneven. Training on 2D triangle and square datasets transfers much of the house's translational dynamics but not its rotation, and more complex shapes such as the pentagon remain challenging for the current setup, which may require a generalist model trained on diverse data. Finally, our theory uses a linear feature model, whereas the neural predictor is nonlinear and history dependent, and contacts can perturb transition branches. Understanding the behavior of JEPA-type models calls for new theoretical models that incorporate nonlinearities as well as action-perturbing effects.

Reproducibility Statement

Every component needed to reproduce our results is specified in the paper or its appendices. The accompanying code is available in an anonymized repository:https://anonymous.4open.science/r/sg-jepa-55DD/. It contains the world-model, probe, and diffusion-policy training code, the MuJoCo dataset generators, the configuration files for every reported run, stage-by-stage example workflows, and a pinned environment specification. Section 2 defines the SG-JEPA architecture, the discounted K-step rollout objective, and the SIGReg term, and Appendix C gives the complete encoder, predictor, and action-encoder configurations, the hybrid Muon/AdamW recipe with all learning rates and schedules, and the matched configurations used for the Original LeWM and DINO-WM baselines. Appendix B documents the generation of all eight MuJoCo environments, including action schemas, the training gravity distributions, the held-out gravity grids, and the episode-level data splits. Appendix C.2 specifies the state probes and the gravity-conditioned Diffusion Policies, including the receding-horizon parameters and the success criterion for each control task. The hyperparameter choices quoted in the main text are supported by the sweeps in Appendix D, covering learning rate, optimizer, SIGReg coefficient, history length, training rollout horizon, and rollout discount. All assumptions, statements, and full proofs for the linear feature model of Section 4, including the coverage bound (Eq. 3) and the composition results, are given in Appendix H. We also state the limits of our protocol: each world model and predictor is trained with a single seed, so the prediction comparisons average over held-out episodes rather than over training runs, whereas each control result averages five policy-rollout seeds with fixed checkpoints and episode identifiers.

AI Usage Statement

We disclose the use of AI assistance in preparing this submission. Large language models were used as a general-purpose coding and writing aid: they assisted primarily with synthetic dataset-generation scripts and plotting code, and with editing the manuscript for clarity. The research question, the SG-JEPA method, the experimental design, the theoretical model and its proofs, and the interpretation of the results are the authors' own work. We have reviewed all AI-assisted work: AI-assisted code was read and checked against simulator ground truth by the authors, and AI-assisted text was reviewed and revised by the authors. The authors take full responsibility for the entire contents of this paper, including every part prepared with AI assistance.

References

Mahmoud Assran, Quentin Duval, Ishan Misra, Piotr Bojanowski, Pascal Vincent, Michael Rabbat, Yann LeCun, and Nicolas Ballas. Self-supervised learning from images with a joint-embedding predictive architecture. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 15619–15629, 2023. doi: 10.1109/CVPR52729.2023.01499.

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

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

Philip J. Ball, Cong Lu, Jack Parker-Holder, and Stephen Roberts. Augmented world models facilitate zero-shot dynamics generalization from a single offline environment. In *Proceedings of the 38th International Conference on Machine Learning*, volume 139 of *Proceedings of Machine Learning Research*, pp. 619–629. PMLR, 2021.

Adrien Bardes, Quentin Garrido, Jean Ponce, Xinlei Chen, Michael Rabbat, Yann LeCun, Mido Assran, and Nicolas Ballas. Revisiting feature prediction for learning visual representations from video. *Transactions on Machine Learning Research*, 2024.

Daniel Bear, Elias Wang, Damian Mrowca, Felix Binder, Hsiao-Yu Tung, R. T. Pramod, Cameron Holdaway, Sirui Tao, Kevin Smith, Fan-Yun Sun, Fei-Fei Li, Nancy Kanwisher, Joshua Tenenbaum, Daniel Yamins, and Judith Fan. Physion: Evaluating physical prediction from vision in humans and machines. In *Proceedings of the Neural Information Processing Systems Track on Datasets and Benchmarks*, volume 1, 2021.

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

Cheng Chi, Siyuan Feng, Yilun Du, Zhenjia Xu, Eric Cousineau, Benjamin Burchfiel, and Shuran Song. Diffusion policy: Visuomotor policy learning via action diffusion. In *Proceedings of Robotics: Science and Systems*, 2023.

Kyunghyun Cho, Bart van Merriënboer, Caglar Gulcehre, Dzmitry Bahdanau, Fethi Bougares, Holger Schwenk, and Yoshua Bengio. Learning phrase representations using RNN encoder–decoder for statistical machine translation. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing*, pp. 1724–1734. Association for Computational Linguistics, 2014. doi: 10.3115/v1/D14-1179.

Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, Jakob Uszkoreit, and Neil Houlsby. An image is worth 16x16 words: Transformers for image recognition at scale. In *International Conference on Learning Representations*, 2021.

Quentin Garrido, Nicolas Ballas, Mahmoud Assran, Adrien Bardes, Laurent Najman, Michael Rabbat, Emmanuel Dupoux, and Yann LeCun. Intuitive physics understanding emerges from self-supervised pretraining on natural videos. *arXiv preprint arXiv:2502.11831*, 2025.

Christopher Grimm, André Barreto, Satinder Singh, and David Silver. The value equivalence principle for model-based reinforcement learning. In *Advances in Neural Information Processing Systems*, volume 33, 2020.

Albert Gu and Tri Dao. Mamba: Linear-time sequence modeling with selective state spaces. In *First Conference on Language Modeling*, 2024.

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, Jurgis Pasukonis, Jimmy Ba, and Timothy Lillicrap. Mastering diverse control tasks through world models. *Nature*, 640(8059):647–653, 2025. doi: 10.1038/s41586-025-08744-2.

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

W. P. M. H. Heemels, Bart De Schutter, and Alberto Bemporad. Equivalence of hybrid dynamical models. *Automatica*, 37(7):1085–1091, 2001. doi: 10.1016/S0005-1098(01)00059-0.

Keller Jordan, Yuchen Jin, Vlado Boza, Jiacheng You, Franz Cesista, Laker Newhouse, and Jeremy Bernstein. Muon: An optimizer for hidden layers in neural networks. https://kellerjordan.github.io/posts/muon/, 2024. Blog post.

Taylor W. Killian, Samuel Daulton, George Konidaris, and Finale Doshi-Velez. Robust and efficient transfer learning with hidden parameter markov decision processes. In *Advances in Neural Information Processing Systems*, volume 30, pp. 6250–6261, 2017.

Matthieu Kirchmeyer, Yuan Yin, Jeremie Dona, Nicolas Baskiotis, Alain Rakotomamonjy, and Patrick Gallinari. Generalizing to new physical systems via context-informed dynamics model. In *Proceedings of the 39th International Conference on Machine Learning*, volume 162 of *Proceedings of Machine Learning Research*, pp. 11283–11301. PMLR, 2022.

Ashish Kumar, Zipeng Fu, Deepak Pathak, and Jitendra Malik. RMA: Rapid motor adaptation for legged robots. In *Proceedings of Robotics: Science and Systems*, 2021. doi: 10.15607/RSS.2021.XVII.011.

Nathan Lambert, Brandon Amos, Omry Yadan, and Roberto Calandra. Objective mismatch in model-based reinforcement learning. In *Proceedings of the 2nd Conference on Learning for Dynamics and Control*, volume 120 of *Proceedings of Machine Learning Research*, pp. 761–770. PMLR, 2020.

Nathan O. Lambert, Albert Wilcox, Howard Zhang, Kristofer S. J. Pister, and Roberto Calandra. Learning accurate long-term dynamics for model-based reinforcement learning. In *2021 60th IEEE Conference on Decision and Control*, pp. 2880–2887, 2021. doi: 10.1109/CDC45484.2021.9683134.

JB Lanier, Kyungmin Kim, Armin Karamzade, Yifei Liu, Ankita Sinha, Kat He, Davide Corsi, and Roy Fox. Adapting world models with latent-state dynamics residuals. In *Proceedings of the 8th Annual Learning for Dynamics and Control Conference*, volume 331 of *Proceedings of Machine Learning Research*, pp. 117–144. PMLR, 2026.

Yann LeCun. A path towards autonomous machine intelligence. OpenReview, 2022. Version 0.9.2.

Kimin Lee, Younggyo Seo, Seunghyun Lee, Honglak Lee, and Jinwoo Shin. Context-aware dynamics model for generalization in model-based reinforcement learning. In *Proceedings of the 37th International Conference on Machine Learning*, volume 119 of *Proceedings of Machine Learning Research*, pp. 5757–5766. PMLR, 2020.

Yunzhu Li, Toru Lin, Kexin Yi, Daniel Bear, Daniel Yamins, Jiajun Wu, Joshua Tenenbaum, and Antonio Torralba. Visual grounding of learned physical models. In *Proceedings of the 37th International Conference on Machine Learning*, volume 119 of *Proceedings of Machine Learning Research*, pp. 5927–5936. PMLR, 2020.

Jingyuan Liu, Jianlin Su, Xingcheng Yao, Zhejun Jiang, Guokun Lai, Yulun Du, Yidao Qin, Weixin Xu, Enzhe Lu, Junjie Yan, Yanru Chen, Huabin Zheng, Yibo Liu, Shaowei Liu, Bohong Yin, Weiran He, Han Zhu, Yuzhi Wang, Jianzhou Wang, Mengnan Dong, Zheng Zhang, Yongsheng Kang, Hao Zhang, Xinran Xu, Yutao Zhang, Yuxin Wu, Xinyu Zhou, and Zhilin Yang. Muon is scalable for LLM training. *arXiv preprint arXiv:2502.16982*, 2025.

Ilya Loshchilov and Frank Hutter. Decoupled weight decay regularization. In *International Conference on Learning Representations*, 2019.

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.

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.

Arjun Majumdar, Karmesh Yadav, Sergio Arnaud, Jason Ma, Claire Chen, Sneha Silwal, Aryan Jain, Vincent-Pierre Berges, Tingfan Wu, Jay Vakil, Pieter Abbeel, Jitendra Malik, Dhruv Batra, Yixin Lin, Oleksandr Maksymets, Aravind Rajeswaran, and Franziska Meier. Where are we in the search for an artificial visual cortex for embodied intelligence? In *Advances in Neural Information Processing Systems*, volume 36, 2023. doi: 10.52202/075280-0031.

Saman Motamed, Laura Culp, Kevin Swersky, Priyank Jaini, and Robert Geirhos. Do generative video models understand physical principles? In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, pp. 948–958, 2026. doi: 10.1109/WACV61042.2026.00099.

Suraj Nair, Aravind Rajeswaran, Vikash Kumar, Chelsea Finn, and Abhinav Gupta. R3M: A universal visual representation for robot manipulation. In *Conference on Robot Learning*, volume 205 of *Proceedings of Machine Learning Research*, pp. 892–909, 2022.

Maxime Oquab, Timothée Darcet, Théo Moutakanni, Huy V. Vo, Marc Szafraniec, Vasil Khalidov, Pierre Fernandez, Daniel Haziza, Francisco Massa, Alaaeldin El-Nouby, Mido Assran, Nicolas Ballas, Wojciech Galuba, Russell Howes, Po-Yao Huang, Shang-Wen Li, Ishan Misra, Michael Rabbat, Vasu Sharma, Gabriel Synnaeve, Hu Xu, Hervé Jégou, Julien Mairal, Patrick Labatut, Armand Joulin, and Piotr Bojanowski. DINOv2: Learning robust visual features without supervision. *Transactions on Machine Learning Research*, 2024.

William Peebles and Saining Xie. Scalable diffusion models with transformers. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 4195–4205, 2023. doi: 10.1109/ICCV51070.2023.00387.

Ronan Riochet, Mario Ynocente Castro, Mathieu Bernard, Adam Lerer, Rob Fergus, Véronique Izard, and Emmanuel Dupoux. IntPhys 2019: A benchmark for visual intuitive physics understanding. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 44(9):5016–5025, 2022. doi: 10.1109/TPAMI.2021.3083839.

Alvaro Sanchez-Gonzalez, Jonathan Godwin, Tobias Pfaff, Rex Ying, Jure Leskovec, and Peter W. Battaglia. Learning to simulate complex physics with graph networks. In *Proceedings of the 37th International Conference on Machine Learning*, volume 119 of *Proceedings of Machine Learning Research*, pp. 8459–8468. PMLR, 2020.

Julian Schrittwieser, Ioannis Antonoglou, Thomas Hubert, Karen Simonyan, Laurent Sifre, Simon Schmitt, Arthur Guez, Edward Lockhart, Demis Hassabis, Thore Graepel, Timothy Lillicrap, and David Silver. Mastering Atari, Go, chess and shogi by planning with a learned model. *Nature*, 588: 604–609, 2020. doi: 10.1038/s41586-020-03051-4.

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.

Erik Talvitie. Model regularization for stable sample rollouts. In *Proceedings of the Thirtieth Conference on Uncertainty in Artificial Intelligence*, pp. 780–789, 2014.

Kaizhen Tan, Xin Xu, Siru Tao, Yixiao Li, Hanzhe Hong, Yang Feng, and Heqing Du. What can latent world models know? physical parameter identifiability in multimodal predictive representations. *arXiv preprint arXiv:2607.27017*, 2026.

Josh Tobin, Rachel Fong, Alex Ray, Jonas Schneider, Wojciech Zaremba, and Pieter Abbeel. Domain randomization for transferring deep neural networks from simulation to the real world. In *2017 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pp. 23–30, 2017. doi: 10.1109/IROS.2017.8202133.

Emanuel Todorov, Tom Erez, and Yuval Tassa. MuJoCo: A physics engine for model-based control. In *2012 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pp. 5026–5033, 2012. doi: 10.1109/IROS.2012.6386109.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. Attention is all you need. In *Advances in Neural Information Processing Systems*, volume 30, 2017.

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

Junjie Wang, Qichao Zhang, Yao Mu, Dong Li, Dongbin Zhao, Yuzheng Zhuang, Ping Luo, Bin Wang, and Jianye Hao. Prototypical context-aware dynamics for generalization in visual control with model-based reinforcement learning. *IEEE Transactions on Industrial Informatics*, 20(9): 10717–10727, 2024. doi: 10.1109/TII.2024.3396525.

Shuai Wang, Yaxin Feng, Xuekun Jiang, Shihan Tian, Ningyu Yan, Xing Shen, Chaoyang Lyu, Hui Wang, Yunsong Zhou, Hanqing Wang, Jiangmiao Pang, Yang Xiang, Xing Gao, Chunhua Shen, and Weinan Zhang. GAUGE: A measurement-grounded benchmark for physical fidelity in simulation engines and video world models. *arXiv preprint arXiv:2608.05948*, 2026.

Matthew O. Williams, Ioannis G. Kevrekidis, and Clarence W. Rowley. A data-driven approximation of the Koopman operator: Extending dynamic mode decomposition. *Journal of Nonlinear Science*, 25(6):1307–1346, 2015. doi: 10.1007/s00332-015-9258-5.

Haotian Xue, Yipu Chen, Liqian Ma, Zelin Zhao, Lama Moukheiber, Yuchen Zhu, and Yongxin Chen. ACWM-Phys: Investigating generalized physical interaction in action-conditioned video world models. *arXiv preprint arXiv:2605.08567*, 2026.

Wenhao Yu, Jie Tan, C. Karen Liu, and Greg Turk. Preparing for the unknown: Learning a universal policy with online system identification. In *Proceedings of Robotics: Science and Systems*, 2017. doi: 10.15607/RSS.2017.XIII.048.

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.

CONTENTS

A Additional related work **16**

B Datasets **16**

C Model and training details **19**

C.1 World Model 19

C.2 Downstream model 20

D Ablation studies **20**

D.1 Optimizer and hyperparameter choices 20

D.2 History length and rollout horizon ablations 24

D.3 Rollout discount 24

D.4 Counterfactual gravity conditioning 26

D.5 Where does GRU advantage come from 26

D.6 Performance on conventional control tasks 29

E 2D datasets full results **29**

E.1 Gravity-conditioned rollouts 29

E.2 Shape generalization 31

F 3D datasets full results **33**

F.1 Approach-Ball results 33

F.2 Arm-Catcher-Ball control results 33

F.3 Franka paddle-to-basket results 34

F.4 Arm-Paddle-Ball results 35

G Sparse-Gravity Post-Training Ablation **36**

H Theory of prediction under changing physical laws **39**

H.1 Setup: the model 40

H.2 Local prediction error under a fixed gravity 41

H.3 How gravity coverage controls test error 42

H.4 Local errors under repeated composition 45

H.5 Why multi-step composition can change the learned representation 48

H.6 Relating the theory to the history-based predictor 51

H.7 Regime changes and contact 53

H.8 Supplementary calculation: SIGReg and its limits 55

A Additional related work

This appendix expands Section 1.

Latent world models and physical prediction. Joint-embedding predictive architectures learn to predict features rather than pixels (LeCun, 2022; Assran et al., 2023; Bardes et al., 2024; Assran et al., 2025). DINO-WM fits dynamics over a frozen DINOv2 encoder (Zhou et al., 2025; Oquab et al., 2024), while LeWM trains its encoder and predictor jointly with the SIGReg regularizer from LeJEPA (Balestriero & LeCun, 2025; Maes et al., 2026). Such latent models can also plan from reward-free offline data (Sobal et al., 2025); reconstruction-based world models retain a pixel or observation decoder (Ha & Schmidhuber, 2018; Hafner et al., 2019; 2025; Hansen et al., 2024). Physical benchmarks provide a complementary test. IntPhys and Physion evaluate physical events or human judgments (Riochet et al., 2022; Bear et al., 2021), whereas Physics-IQ and GAUGE measure trajectories and physical quantities (Motamed et al., 2026; Wang et al., 2026). Other work studies intuitive physics in self-supervised video representations (Garrido et al., 2025), structured particle simulators (Sanchez-Gonzalez et al., 2020), visual inference of physical properties (Li et al., 2020), and prediction under held-out physical configurations (Xue et al., 2026; Tan et al., 2026).

Generalization across physical systems. Hidden-parameter MDPs represent related environments with a low-dimensional dynamics variable (Killian et al., 2017). CaDM and CoDA infer this context from trajectories (Lee et al., 2020; Kirchmeyer et al., 2022), and ProtoCAD extends context-conditioned dynamics to high-dimensional visual observations (Wang et al., 2024). Universal policies, rapid motor adaptation, and augmented world models also infer or adapt to changing dynamics (Yu et al., 2017; Kumar et al., 2021; Ball et al., 2021); residual latent dynamics provide a related approach to sim-to-real adaptation (Lanier et al., 2026). Domain randomization instead trains across parameter variations without exposing the parameter to the policy (Tobin et al., 2017). SG-JEPA removes system identification from the experiment: the scene and control interface remain fixed, gravity is given to the model, and only that scalar varies. The evaluation therefore isolates interpolation and extrapolation along a known physical coordinate.

Rollout training and downstream use. A one-step model accumulates error when its predictions become future inputs (Bengio et al., 2015). Existing remedies feed predictions back during training (Talvitie, 2014), correct rollout states with training data (Venkatraman et al., 2015), supervise several latent horizons (Hafner et al., 2019), or optimize long-horizon accuracy directly (Lambert et al., 2021). SG-JEPA backpropagates its rollout loss through both predictor and encoder, and the frozen-encoder crossover separates their contributions across recurrent, state-space, and attention predictors (Cho et al., 2014; Vaswani et al., 2017; Gu & Dao, 2024). Because model loss need not track control performance (Lambert et al., 2020), related work has used value-equivalent objectives (Grimm et al., 2020; Schrittwieser et al., 2020) and frozen representations for imitation learning (Nair et al., 2022; Majumdar et al., 2023; Chi et al., 2023). We likewise report physical prediction and downstream policy performance, while keeping the data and policy training fixed within each controlled comparison. The accompanying analysis uses lifted linear dynamics and a piecewise-affine contact model (Williams et al., 2015; Lusch et al., 2018; Heemels et al., 2001); these are surrogate analyses, not claims that the learned predictors are linear.

B Datasets

We evaluate eight datasets generated with MuJoCo (Todorov et al., 2012): four planar rigid-body prediction tasks, one 3D projectile prediction task, and three 3D robotic control tasks. Table 1 summarizes their observation, action, and state records.

Planar rigid-body prediction. The four 2D datasets contain one red rigid body, a right triangle, square, pentagon, or "house" (concatenation of the right triangle and the square along the leg), moving and colliding inside a planar box. We randomize its initial pose and apply a single planar impulse. The eight-dimensional state is $[x, z, v_x, v_z, \vartheta, \omega, x_{\text{anchor}}, z_{\text{anchor}}]$. The physics record also stores the object mass and size, shape identifier, and box dimensions.

3D prediction and control. *Approach Ball* isolates projectile prediction: a ball is launched with randomized initial velocity and travels freely without actions. *Arm Catcher Ball* and *Arm Paddle Ball* use a Unitree Z1 arm. The catcher task moves a catcher to intercept the bouncing ball, while the paddle task controls position and tilt to sustain repeated bounces. Their per-frame records include the commanded and realized catcher or paddle state, contact flags, arm joint positions and velocities, and task success. *Franka Paddle-to-Basket* instead uses a Franka arm with a blue paddle attached to strike an incoming ball toward a basket. Success requires a valid blade contact followed by the ball entering the basket. In addition to the ball, paddle, contact, physics, and arm records, this dataset stores basket geometry, task events, a strike diagnostic, and a failure label. The scripted outcome cycle contains 70% successes and 5% from each of six informative failure modes.

Table 1: Dataset summary. Each episode consists of 64 frames at 16 Hz. State records contain the task-relevant object and robot states used for supervision and evaluation.

Dataset	Role	RGB resolution	Action record	Principal state record
Right Triangle	Prediction	128^2	(J_x, J_z, g)	Rigid body (8)
Square	Prediction	128^2	(J_x, J_z, g)	Rigid body (8)
Pentagon	Prediction	128^2	(J_x, J_z, g)	Rigid body (8)
House	Prediction	128^2	(J_x, J_z, g)	Rigid body (8)
Approach Ball	Prediction	256^2	g	Ball (16)
Arm Catcher Ball	Control	256^2	$(g, \Delta x, \Delta y, \Delta z)$	Ball (16), catcher (9), Z1 arm (18)
Arm Paddle Ball	Control	256^2	$(g, \Delta x, \Delta y, \Delta z, \phi, \theta)$	Ball (16), paddle (9), Z1 arm (18)
Franka Paddle-to-Basket	Control	256^2	$(g, \Delta x, \Delta y, \Delta z, \phi, \theta)$	Ball (16), paddle (9), Panda arm (21), basket (6)

Gravity sampling and input encoding. The gravity g is constant within each episode. For the 2D planar tasks, training gravity follows $g = \max(\mathcal{N}(4, 0.5^2), 0.1)$ with 8000 episodes each, and the held-out test split uses the 25-point grid $\{-2, -1.5, \ldots, 10\}$ with 200 episodes per value. For 3D datasets, *Approach Ball*, *Arm Paddle Ball* and *Franka Paddle-to-Basket* again samples 8000 episodes from $g = \max(\mathcal{N}(9.8, 2^2), 0)$ for training; and 25-value held-out test set $\{0, 1, \ldots, 20, 0.62, 1.63, 3.72, 8.87\}$, with the last 4 values being the gravity values in Pluto, Moon, Mars and Venus respectively. *Arm Catcher Ball* uses $g = \max(\mathcal{N}(4, 0.5^2), 0.1)$ and the 23-value grid $\{-1, -0.5, \ldots, 10\}$.

Gravity is stored in physical units in the physics record and as an action coordinate. The planar action is (J_x, J_z, g), where the two impulse channels are nonzero only at $t = 0$. *Approach Ball* receives only g. The arm tasks append Cartesian catcher or paddle commands from a scripted expert, and *Arm Paddle Ball* and *Franka Paddle-to-Basket* append two paddle-orientation commands. Before a learned component receives gravity, we z-score it as $(g - \mu_{g,\text{train}})/\sigma_{g,\text{train}}$. We compute these statistics only on the corresponding training split and keep them fixed for validation, test, and post-training evaluation. Evaluation data never enter the normalization statistics.

Format and storage. Each episode contains 64 frames at 16 Hz over 4 seconds. The 2D videos are 128×128, and the 3D videos are 256×256. Action a_t is aligned with the transition from frame t to frame $t + 1$. The Lance datasets contain one row per frame, including episode and step indices, split identifier, JPEG-encoded RGB, simulator state, aligned action, reward, a compact physics vector, raw gravity, and source-episode index. Controlled tasks add JSON episode metadata and the task-specific records above. Headline 2D plots use fixed named subsets of the larger test split, while some 3D probe diagnostics use separate evaluation manifests; each result states its cohort size.

Figure 6 gives representative trajectories from all eight datasets, grouped by their prediction or control role. Figure 7 compares ground-truth frames with decoded open-loop predictions for one 2D task and two 3D tasks under both in-distribution and out-of-distribution gravity. For the decoded rollouts, we attach a separate online CNN decoder to each world model. The decoder maps the 256-dimensional latent to RGB from an 8×8 feature map, with 256 base channels, a 32-channel minimum, and one residual block. We train it on detached features with RGB MSE, using Muon at 5×10^{-4} for eligible matrix parameters and AdamW at 10^{-4} for the remaining parameters. The reconstruction loss therefore does not update the world model. The decoder is used only for visualization; quantitative prediction results use frozen physical-state probes.

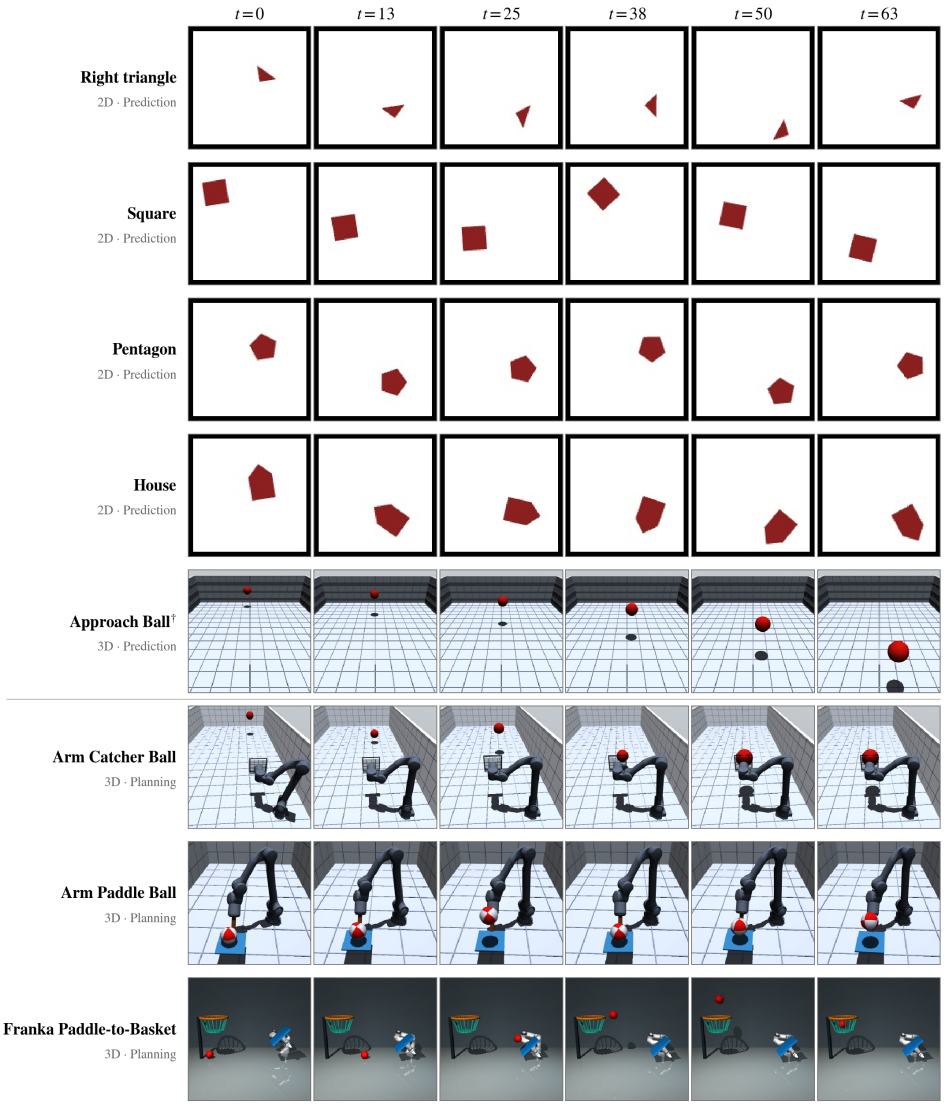


Figure 6: Representative evolution of one episode from each dataset. We show frames $t \in \{0, 13, 25, 38, 50, 63\}$. Rows above the divider are passive prediction environments; rows below it require arm control and are used for control.

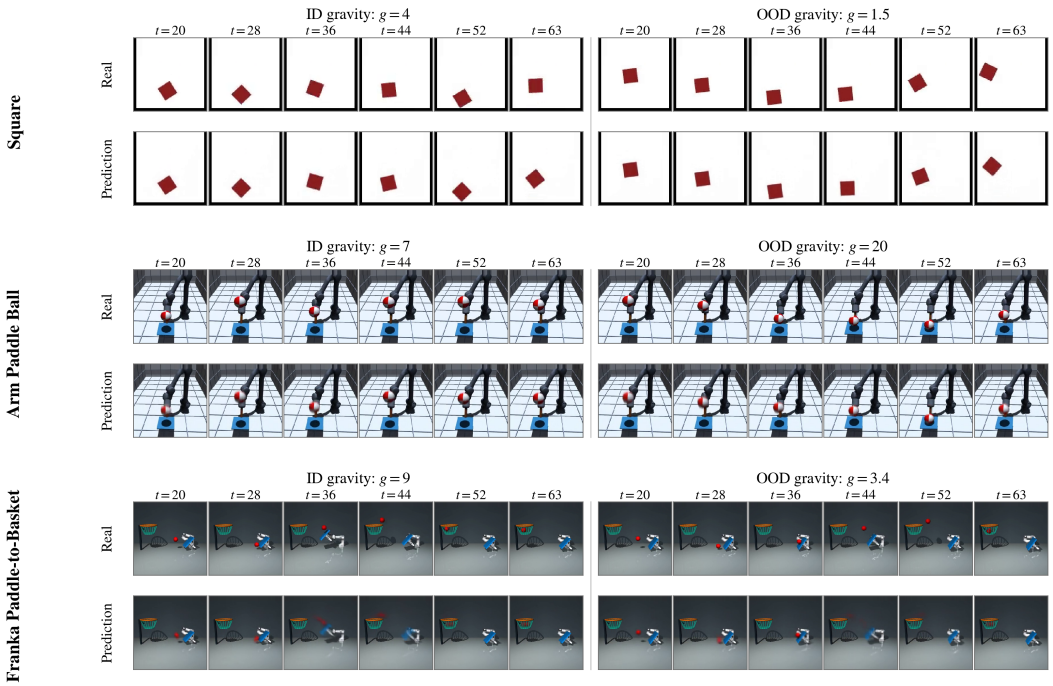


Figure 7: Decoded open-loop rollouts under in-distribution (ID) and out-of-distribution (OOD) gravity. For each dataset, the upper row shows ground-truth and the lower row decodes the autoregressively predicted latents. The model receives 20 context frames and then rolls out through $t = 63$.

C Model and Training Details

C.1 World Model

Section 2 gives the model architecture and training objective; in this section, we elaborate on the implementation details needed to reproduce the reported runs. The ViT-Tiny CLS token is projected from 192 to 256 dimensions. The GRU and SSM predictors both use three residual layers of width 512, an MLP width of 2048, dropout 0.1, and action concatenation at every layer. As with the other action channels, gravity g is z-scored before entering the action encoder.

During rollout training, each predicted latent is fed back into the predictor. For the next step, the latent and action histories shift forward together, retaining only their most recent H entries. Every final SG-JEPA model uses $H = 20$, a five-step training rollout $K = 5$, discount $\gamma = 0.95$, and trainable target latents without stop-gradient since SIGReg already regulates the latent space and avoids collapse. SIGReg is evaluated on encoded latents only, using 1024 random projections and 17 knots. The coefficient is selected separately for each task family, as summarized in Table 2. All models are trained for 20 epochs with batchsize 64. Ablation studies for these hyperparameters are presented in the next sections.

All rows use the same hybrid optimizer. Every trainable two-dimensional parameter tensor is optimized by Muon at 10^{-4}; the remaining parameters use AdamW at 5×10^{-5} with weight decay 10^{-3}. Muon uses Nesterov momentum 0.95, five Newton–Schulz steps, and the adjustment with scale 0.4 Liu et al. (2025).

Baseline configurations. Original LeWM retains the same ViT-Tiny encoder, 256-dimensional projector, action preprocessing, and $H = 20$ context, but uses a causal Transformer with six layers, width 256, 16 attention heads, MLP width 2048, and dropout 0.1. It is trained with the teacher-forced one-step latent MSE and SIGReg coefficient 0.09. These runs use the same hybrid Muon–AdamW recipe above and fixed epoch-20 checkpoints.

Table 2: SG-JEPA configurations used in the main textheadline experiments. All reported models use the fixed epoch-20 checkpoint.

Task family	**Image / patch**	**Predictor**	$\boldsymbol{\lambda_{\text{sig}}}$
Planar rigid bodies	$128^2/8$	GRU, SSM	0.72
Approach Ball	$256^2/16$	GRU, SSM	0.18
Arm Catcher Ball	$256^2/16$	GRU	0.18
Arm Paddle Ball	$256^2/16$	GRU	0.72
Franka Paddle-to-Basket	$256^2/16$	GRU	0.09

DINO-WM freezes a pretrained DINOv2 ViT-S/14 encoder and extracts a grid of 64 normalized patch tokens, each 384-dimensional, from 112×112 inputs. Its action-conditioned, frame-causal Transformer uses $H = 20$, six layers, 16 attention heads, MLP width 2048, dropout 0.1, and a 10-dimensional action embedding. The predictor is trained by teacher forcing with one-step MSE to the next frozen patch-token grid. The reported DINO-WM predictors are trained for 20 epochs with AdamW at 5×10^{-4}, weight decay 10^{-2}, batch size 32.

C.2 Downstream model

State probes. For each frozen source representation, we fit a separate MLP probe. The probe layer-normalizes the concatenated latent window, followed by two GELU layers of widths 512 and 256 with dropout 0.05 and a linear readout. We z-score each target coordinate using statistics from the probe-training split and optimize the probe with AdamW at 10^{-3}, weight decay 10^{-5}, and batch size 256 for at most 50 epochs, retaining the checkpoint with the lowest validation NMSE. Probes are trained only on encoded ground-truth sequences and are then frozen before being applied to predicted rollouts.

Diffusion policies. We train a separate policy for every frozen encoder and control task. In addition to the latent-history GRU described in Section 2.3, the noise network is a conditional 1D U-Net with channel widths $256, 512, 1024$, kernel size 5, eight GroupNorm groups, and a 256-dimensional diffusion-step embedding. At diffusion step τ, an action block $\mathbf{u} = u_{t:t+A-1}$ of A steps from the training trajectories is corrupted as $\mathbf{u}^{(\tau)} = \sqrt{\bar{\alpha}_\tau}\,\mathbf{u} + \sqrt{1-\bar{\alpha}_\tau}\,\epsilon$, where $\epsilon \sim \mathcal{N}(0, I)$ and $\bar{\alpha}_\tau$ is the cumulative coefficient of the cosine noise schedule. The U-Net predicts the added noise from $\mathbf{u}^{(\tau)}$, τ, and the conditioning vector ξ of Section 2.3, minimizing

$$\mathcal{L}_{\text{DP}} = \mathbb{E}_{\tau,\epsilon}\left[\left\|\epsilon_\omega(\mathbf{u}^{(\tau)}, \tau \mid \xi) - \epsilon\right\|_2^2\right]. \tag{5}$$

Each policy is trained for 300k updates with batch size 256 using AdamW at 10^{-4}, weight decay 10^{-6}. The learning rate warms up linearly for 500 updates and then decays with a cosine schedule. The visual encoder remains frozen throughout. In particular, the DINO-WM policy mean-pools frozen patch tokens, and neither it nor the SG-JEPA policies query a world-model predictor online.

D Ablation studies

D.1 Optimizer and hyperparameter choices

D.1.1 Learning rate sweep

We run learning rate sweep on the 2D 128×128 square dataset using a Transformer predictor. The 8,000 source episodes are partitioned by episode into 7,200 training and 800 validation episodes; panels (d)–(f) use extra held-out test split. We vary the AdamW learning rate over $\{10^{-5}, 3 \times$

$10^{-5}, 5\times10^{-5}, 10^{-4}, 2\times10^{-4}, 5\times10^{-4}\}$, with weight decay 10^{-3}, batch size 64, $H = 20$, $K = 5$, $\gamma = 0.95$, and $\lambda_{\text{SIG}} = 0.72$. Each run trains for at most 20 epochs. Early stopping monitors the validation objective with patience three, and we retain the checkpoint with the lowest validation objective. We then fit temporal-window-4 linear and MLP probes to each checkpoint.

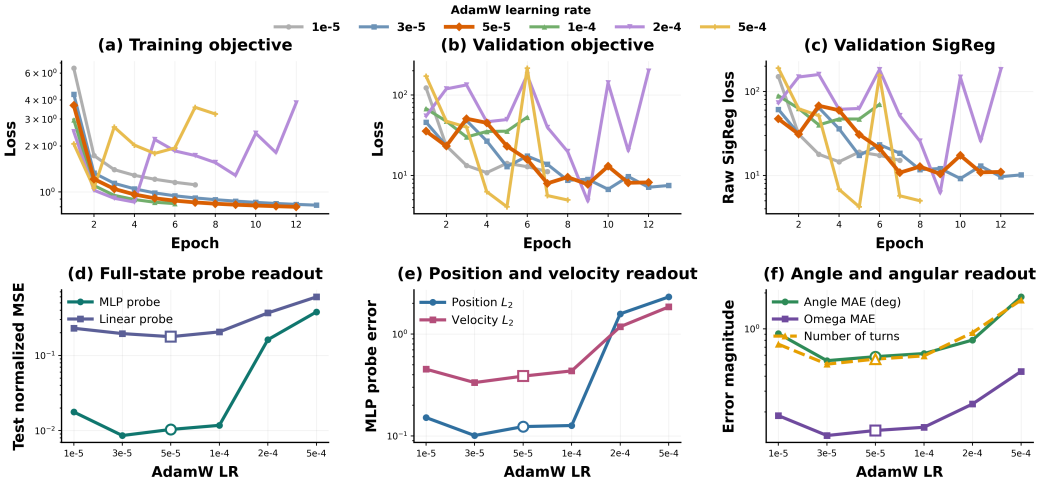


Figure 8: **AdamW learning-rate sweep on 2D square dataset**

Results The lowest validation objective occurs at 5×10^{-4}, but its poor probe readout shows that the objective alone is not a reliable measure of representation quality. The 5×10^{-5} setting nearly matches the best MLP-probe validation NMSE and gives the best linear-probe NMSE; the held-out test results in Figure 8 show the same trend. We therefore use 5×10^{-5} for the AdamW parameters.

D.1.2 Muon versus AdamW

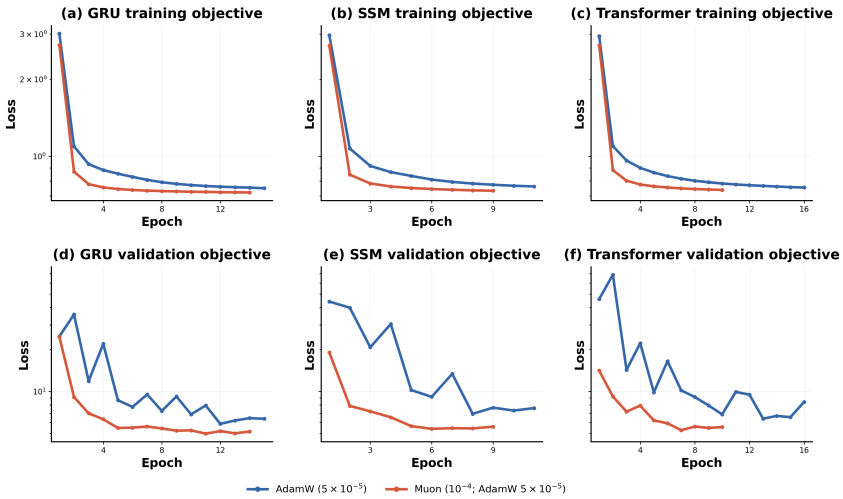


Figure 9: **AdamW versus hybrid Muon/AdamW** Panels (a-c) show training objectives and panels (d-f) show validation objectives for the GRU, SSM, and Transformer predictors. Lower is better.

In Fig. 9, we compare AdamW with the hybrid Muon/AdamW on the 128×128 square dataset. For each of the GRU, SSM, and Transformer predictors, both conditions use a 256-dimensional latent, $H = 20$, $K = 5$, $\gamma = 0.95$, $\lambda_{\text{SIG}} = 0.72$, batch size 64. Training is capped at 20 epochs, with early stopping after three epochs without validation improvement. AdamW uses 5×10^{-5} for all parameters. The hybrid recipe assigns every two-dimensional parameter tensor to Muon at 10^{-4}

and the remaining parameters to AdamW at 5×10^{-5}. Muon uses Nesterov momentum 0.95, five Newton-Schulz iterations, and per-matrix RMS-matching multiplier $0.4\sqrt{\max(A, B)}$.

Across all three predictors, Muon lowers the training objective faster and reaches a lower best validation objective: 4.990 versus 5.864 for the GRU, 5.403 versus 6.922 for the SSM, and 5.278 versus 6.384 for the Transformer. These are relative reductions of 14.9%, 21.9%, and 17.3%, respectively. The selected checkpoints also occur earlier, at epochs 11, 6, and 7 instead of 12, 8, and 13.

D.1.3 SIGREG COEFFICIENT SWEEP

In this case we vary $\lambda_{\text{SIG}} \in \{0.09, 0.18, 0.36, 0.72, 1.44\}$ on 2D square dataset. Rollout metrics and decoded examples use a 5,000-episode selection cohort spanning 25 gravity values from -2 to 10; effective rank uses a 2,000-episode well-spread subset. We select SIGReg coefficient based on the downstream rollout performance.

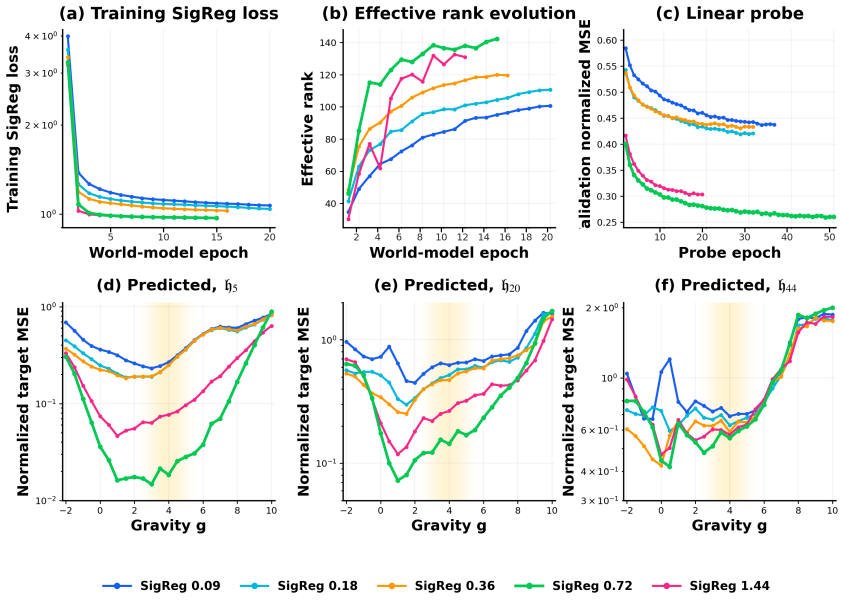


Figure 10: **SIGReg coefficient sweep.** (a) Raw training SIGReg loss; (b) Effective rank on the same 2,000 episodes at every checkpoint; (c) Linear-probe validation NMSE; and (d–f) MLP-probe NMSE after 5, 20, and 44 autoregressive steps across $g \in [-2, 10]$. Panels (c–f) use the same epoch checkpoints. The yellow shade in the background indicates training distribution. Lower is better except in (b).

We note that raw SIGReg loss measures agreement with a standard Gaussian under random one-dimensional projections, not dimensional collapse. We therefore also report the effective rank of the final-context latent covariance. For $z_i \in \mathbb{R}^{256}$ and $N = 2000$,

$$\bar{z} = \frac{1}{N}\sum_{i=1}^{N} z_i, \qquad C = \frac{1}{N-1}\sum_{i=1}^{N}(z_i - \bar{z})(z_i - \bar{z})^\top. \tag{6}$$

Let $\lambda_1 \geq \cdots \geq \lambda_D \geq 0$ denote the eigenvalues of C. We summarize the spectrum by its effective rank,

$$p_j = \frac{\lambda_j}{\sum_k \lambda_k}, \qquad r_{\text{eff}}(C) = \exp\left(-\sum_{j:p_j>0} p_j \log p_j\right). \tag{7}$$

Fig. 10 separates optimization behavior from representation quality. Although 0.72 and 1.44 reach similarly low raw SIGReg losses in panel (a), panels (b) and (c) favor 0.72: the effective rank is 138.0, compared with 131.0 for 1.44, and it gives the lowest linear-probe validation NMSE. Panels

(d) and (e) show that 0.72 also gives the best short- and medium-horizon readout over most of the gravity range. Early rank remains a useful screening signal, with epoch-5 correlations of $\rho_s = -0.90$ and -1.00 against later linear- and MLP-probe NMSE, respectively.

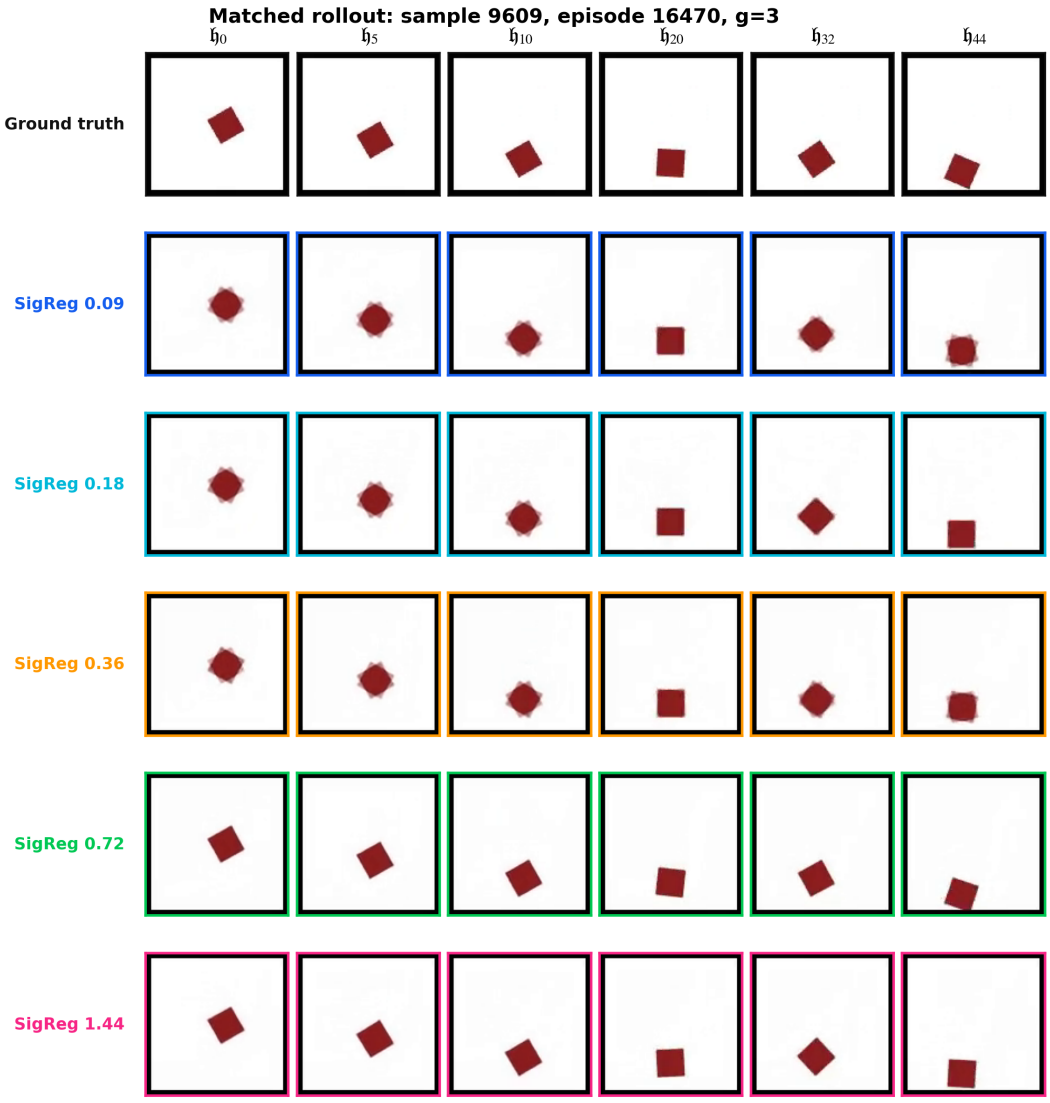


Figure 11: **Qualitative comparison of the SIGReg coefficient.** Matched rollout predictions for one representative square episode at $g = 3$. The top row shows ground-truth frames, and the remaining rows show decoded predicted latents for each value of λ_{SIG}. Columns show the autoregressive rollout horizons 5, 10, 20, 32 and 44. Each prediction row is rendered using the online decoder trained jointly with that model.

Fig. 11 shows the same pattern. The representations obtained with $\lambda_{\text{SIG}} = 0.09, 0.18, 0.36$ decode the square as a rounded or blurred object and accumulate visible pose and position errors. At $\lambda_{\text{SIG}} = 0.72$, the higher-rank representation most clearly preserves corners, orientation, and trajectory over the rollout. Increasing the coefficient to 1.44 retains relatively sharp geometry but introduces larger long-horizon trajectory errors. Finally we select $\lambda_{\text{SIG}} = 0.72$ for our 2D experiments because it gives the best balance of effective rank, frozen-probe accuracy, and rollout quality.

D.2 History length and rollout horizon ablations

To choose a common temporal configuration without running a separate sweep for every dataset, we perform detailed ablations on one planar task (2D Square) and one 3D task (Approach Ball). The 2D study varies history length and predictor depth, whereas the 3D study also varies the SIGReg coefficient (2D we've done in the section above). Both tasks favor a 20-frame history, and Approach Ball favors a five-step training rollout. We therefore use $H = 20$ and $K = 5$ across all datasets for a consistent experimental setup. The detailed results are as follows.

D.2.1 2D square: history length and predictor depth

On the 2D square dataset, we cross $H \in \{5, 10, 20, 30\}$ with predictor depth $L \in \{2, 3, 4\}$ ($H = 40$ causes OOM error on one B200). All runs train for at most 10 epochs. Fig. 12 shows prediction results from trained MLP probes. We see that $H = 20$ gives the best long-horizon accuracy across depths. At this history length, $L = 4$ is numerically best, with position, orientation, and signed-turn errors of 0.745, $13.91°$, and 0.153 turns. The $L = 3$ model is within 2% to 4% on all three metrics, while reducing the predictor from 19.30M to 14.57M parameters. We therefore use $H = 20, L = 3$ as a balance between performance and training cost.

The real-latent controls explain why we do not use $H = 30$. Their orientation error rises from at most $0.66°$ for $H \leq 20$ to about $11.1°$, and rotation error rises from at most 0.052 to 0.446 turns. Thus, the degradation is present in the learned representation before autoregressive error accumulates, even though the $H = 30$ runs attain lower training objectives.

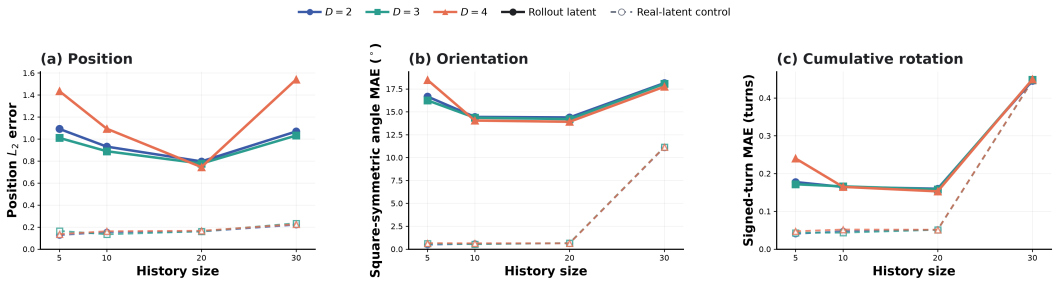


Figure 12: History-length and predictor-depth ablation on 2D quare dataset at a common 32-step rollout horizon. Solid lines use predicted rollout latents; dashed lines use real encoder latents and isolate representation and probe error. Color and marker shape denote predictor depth. Lower is better.

D.2.2 Approach Ball: history and training rollout horizon

In this case we jointly vary $(H, K) \in \{(20, 5), (20, 10), (30, 10), (30, 20)\}$ and $\lambda_{\text{SIG}} \in \{0.18, 0.36, 0.72\}$. For each model, an MLP probe predicts the ball state (x, y, z, v_x, v_y, v_z). Within each (H, K) pair, we select λ_{SIG} by probe-validation NMSE. Table 3 and Fig. 13 show that $(H, K, \lambda_{\text{SIG}}) = (20, 5, 0.18)$ gives the lowest probe-validation NMSE, direct-probe test NMSE, and 20-step rollout NMSE. Its 32-step rollout NMSE is 0.185, within 1.3% of the best value of 0.182 from $(20, 10, 0.18)$, while using 61.3 rather than 73.5 GiB of peak host RAM. The best $H = 30$ variants are both less accurate and more expensive. We therefore use the task-specific setting $H = 20, K = 5$, and $\lambda_{\text{SIG}} = 0.18$ for Approach Ball. The sweep uses one seed, and the resource values describe training-time host memory.

D.3 Rollout discount

We take Arm Catcher Ball dataset and sweep $\gamma \in \{0.90, 0.95, 0.98, 1.00, 1.05\}$. All runs are trained to 20 epochs. We freeze the epoch-20 representation and evaluate an MLP state probe on $\mathfrak{h}_{44}$ predicted latents. Table 4 shows that $\gamma = 0.95$ gives the lowest joint NMSE and velocity error at horizon 44. Its position error is 0.122, close to the best value of 0.118 at $\gamma = 1.00$. We therefore use $\gamma = 0.95$ as the default across datasets.

Training configuration			Normalized MSE ↓			Training resource	
H	K	λ_{SIG}	Probe val.	Probe test	Rollout@32	Decoder frames per batch	Peak host RAM (GiB)
20	**5**	**0.18**	**0.078**	**0.110**	**0.185**	**800**	**61.3**
20	5	0.36	0.118	0.148	0.232	800	69.4
20	5	0.72	0.192	0.225	0.363	800	67.2
20	10	0.18	0.080	0.114	0.182	960	73.5
20	10	0.36	0.092	0.125	0.198	960	80.4
20	10	0.72	0.110	0.143	0.237	960	66.4
30	10	0.18	0.137	0.166	0.347	1280	81.9
30	10	0.36	0.135	0.167	0.236	1280	75.5
30	10	0.72	0.152	0.185	0.263	1280	87.6
30	20	0.18	0.316	0.325	0.392	1600	111.7
30	20	0.36	0.321	0.330	0.401	1600	113.8
30	20	0.72	0.150	0.182	0.260	1600	116.3

Table 3: Approach Ball temporal ablation.

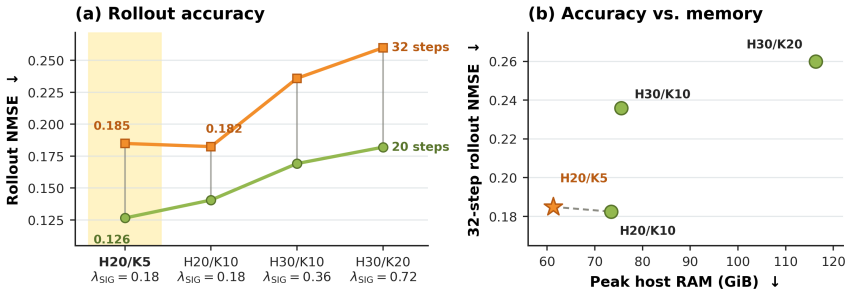


Figure 13: Approach Ball accuracy and resource tradeoffs after selecting λ_{SIG} by probe-validation NMSE within each (H, K) pair. (a) Rollout NMSE at horizons 20 and 32; (b) peak host RAM and decoder frames per microbatch; and (c) the SIGReg sweep for $H = 20, K = 5$. The highlighted configuration is selected. Lower is better.

Table 4: Arm Catcher Ball rollout-discount ablation at horizon 44.

γ	Joint NMSE ↓	Position L_2 ↓	Velocity L_2 ↓
0.90	0.753	0.134	0.398
0.95	**0.600**	0.122	**0.346**
0.98	0.683	0.138	0.351
1.00	0.610	**0.118**	0.354
1.05	0.730	0.145	0.377

D.4 COUNTERFACTUAL GRAVITY CONDITIONING

In this section, we want to test whether the model actually uses the input gravity. We conduct experiments by supplying wrong gravity values in the action, while holding other coordinates fixed. We again train and evaluate on the 2D square dataset. For each $g_{\text{true}} \in \{2, 4, 6\}$, we use the same 200 episodes under $g_{\text{in}} \in \{0, 2, 4, 6, 8\}$. The full-state probe MSE is normalized by the training-set variance of each target coordinate. Let $(\mathfrak{h}_1, \ldots, \mathfrak{h}_7) = (1, 3, 5, 9, 20, 32, 44)$ denote the evaluated forecast steps. We summarize rollout error using the horizon-normalized area under the NMSE curve, computed by the trapezoidal rule:

$$\text{AUC}(g_{\text{in}}) = \frac{1}{\mathfrak{h}_7 - \mathfrak{h}_1} \sum_{i=1}^{6} \frac{\mathfrak{h}_{i+1} - \mathfrak{h}_i}{2} \left[\text{NMSE}_{\mathfrak{h}_i}(g_{\text{in}}) + \text{NMSE}_{\mathfrak{h}_{i+1}}(g_{\text{in}}) \right].$$

We report the paired difference $\Delta\text{AUC} = \text{AUC}(g_{\text{in}}) - \text{AUC}(g_{\text{true}})$; positive values indicate that supplying the mismatched gravity increases rollout error.

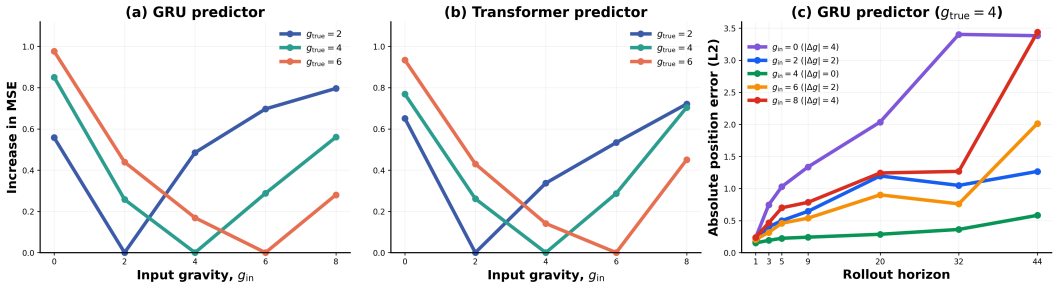


Figure 14: **Counterfactual gravity conditioning** Only the gravity supplied through the action channel is changed. Panels (a) and (b) report the increase in normalized full-state probe NMSE AUC for the GRU and Transformer predictors. Panel (c) reports the mean GRU position L_2 error in the x-z plane across rollout horizons for $g_{\text{true}} = 4$.

Results From the figure we see that for both predictors, the error is minimized when $g_{\text{in}} = g_{\text{true}}$, while mismatched inputs increase the rollout error. Larger gravity mismatches generally produce larger increases, although the response is not symmetric around the true value. At $g_{\text{true}} = 4$, correct conditioning also gives the lowest GRU position error at every reported horizon. These interventions show that the predictor does use the supplied gravity when forecasting dynamics.

D.5 WHERE DOES GRU ADVANTAGE COME FROM

In this section we investigate where the jointly trained GRU advantage comes from: the recurrent predictor itself, the representation it shapes, or a weakness in Transformer conditioning? To this end we conduct three controlled studies. Matching Transformer action conditioning and initialization narrows, but does not close the gap. Under encoder–predictor crossover, the advantage follows the source representation. Reducing history has little source-specific effect over the first five forecasts, but the GRU-trained representation is less sensitive to short context once recursive errors accumulate over a 44-step rollout. Together, these results identify the representation learned via GRU predictor during joint training as the main source of the gain. Detailed results are as follows.

All comparisons use the same seed and the same 5,000 held-out square episodes, with 200 episodes at each of 25 gravity values from -2 to 10. We first average across target dimensions, then across horizons within each episode, and finally across episodes. Subtracting the error obtained from the real encoder latent controls for source-specific probe readability. These intervals measure evaluation-episode variability; all world-model and predictor training uses one seed.

D.5.1 CAN TRANSFORMER CONDITIONING CLOSE THE JOINT-TRAINING GAP?

The first experiment asks whether the joint-training gap comes from how actions are supplied to the Transformer or from the initialization of its conditioning layers. For GRU we concatenate each

latent with its action embedding at the recurrent input. For the Transformer, we test three alternatives: concatenating each latent state with its action once before the first block, reintroducing the action in the attention and MLP paths of every block, or using the action to modulate every block through AdaLN. For AdaLN, we also initialize the conditioning gate to 0, 0.1, or 1, giving five Transformer variants in total. All variants use $H = 20$, $K = 5$, $\gamma = 0.95$, SIGReg coefficient 0.72, batch size 64, 20 epochs. More details are in Table 5

Table 5: Predictor architectures in the action conditioning study.

Method	Action conditioning	Initialization	Depth	Width	MLP width	Parameters
GRU	Action concatenated at recurrent input	residual 0.1	3	512	–	14.57M
AdaLN-0	AdaLN in every block	gate 0	6	256	2048	14.98M
AdaLN-0.1	AdaLN in every block	gate 0.1	6	256	2048	14.98M
AdaLN-1	AdaLN in every block	gate 1	6	256	2048	14.98M
Input concat.	One pre-stack action fusion	residual 0.1	6	256	2776	14.98M
Per-block concat.	Action fusion in attention and MLP paths	residual 0.1	6	256	2304	14.97M

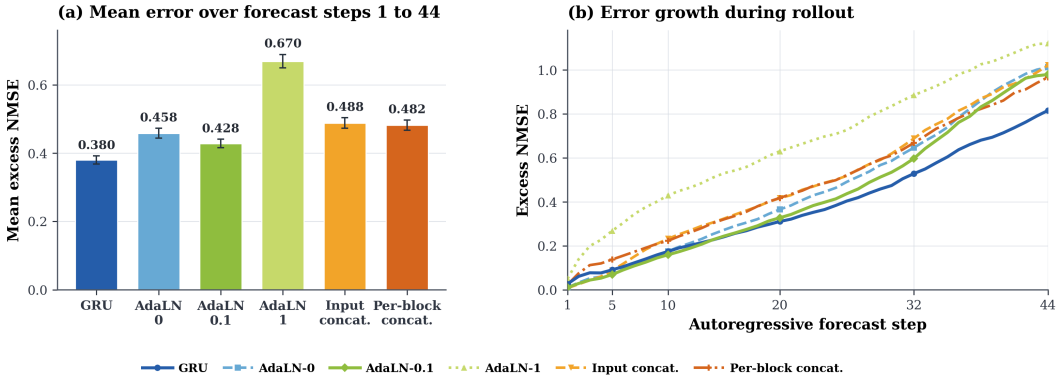


Figure 15: Performance comparison on held-out test set. (a) Mean excess normalized target MSE over forecast steps 1–44; error bars are 95% episode-bootstrap intervals. (b) Excess error by autoregressive forecast step. All models use the same episode IDs and gravity assignments. Lower is better.

Results Fig. 15 shows the results. Initializing the AdaLN gate to 0.1 lowers the Transformer endpoint from 0.458 to 0.428 and closes 38.5% of the gap to the GRU. It remains worse than GRU performance of 0.380, however, with a paired difference of +0.048Å gate of 1.0 degrades the rollout, while input and per-block action concatenation do not improve the aggregate endpoint over AdaLN-0. The best Transformer is worse than the GRU at 22 of 25 gravity values, and per-block concatenation is worse at 25 of 25. Some variants improve individual physical quantities, so the conclusion applies to aggregate rollout error: neither direct action injection nor gate initialization closes the jointly trained GRU gap.

D.5.2 Encoder–predictor crossover: full results

For the second experiment we freeze the encoder from the jointly trained GRU and Transformer with AdaLN, then fit a fresh GRU and Transformer predictor to each latent source. The training setups and hyperparameters are the same except for the predictor. In Section 4 we report the overall crossover result and the main interpretation. Here we provide the full metric breakdown in Table 6. The representation learned jointly with the GRU gives lower overall physical-state error. The new Transformer modestly reduces error on both sources, by -0.107 on the GRU-trained representation and -0.103 on the Transformer-trained representation, yet the GRU-trained representation remains better. This result shows that encoder trained with the GRU predictor produces latents that preserve

dynamics-relevant information that either predictor can use. The lower Euclidean latent MSE of the Transformer-trained representation does not translate into better physical-state prediction, especially for velocity. This contrast confirms that preserving dynamics-relevant state is more informative than raw latent distance.

Table 6: Numerical results for the frozen-representation crossover. The primary endpoint and latent MSE are excess errors averaged over horizons 1 to 44; physical errors use predicted latents at horizon 44.

Frozen source	Fresh predictor	Primary ↓	Latent MSE ↓	Position ↓	Velocity ↓	Orientation (°) ↓	ω ↓
GRU-trained	GRU	1.376	2.247	3.004	3.803	22.88	**2.116**
GRU-trained	Transformer	**1.269**	2.511	3.036	**3.769**	**21.77**	2.139
Transformer-trained	GRU	1.555	**1.638**	**2.828**	5.343	23.15	2.282
Transformer-trained	Transformer	1.453	1.732	2.854	4.849	22.57	2.401

D.5.3 History dependence of the frozen representations

The crossover shows that the advantage follows better representation. We next ask whether the GRU-trained representation is less dependent on long context. For each frozen source, we fit new GRU and Transformer predictors with $m \in \{2, 4, 8, 20\}$ context frames, where m is the number of most recent latents available to the predictor. Each model predicts the same five target frames and is evaluated through 44 autoregressive steps on held-out episodes. The primary comparison shortens the context from $m = 20$ (H20) to $m = 4$ (H4). Following Equation 1, we measure how much the excess physical-state NMSE changes when history is shortened relative to H20. A positive source difference means that the Transformer-trained representation loses more from shorter context. Intervals use 10,000 paired episode bootstrap resamples.

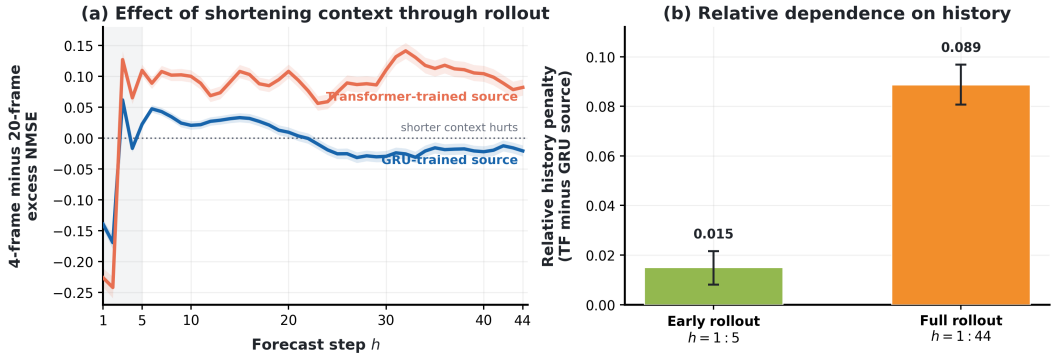


Figure 16: History dependence of the frozen representations. (a) Change in excess physical-state NMSE at each forecast step when context is shortened from H20 to H4, averaged over newly trained GRU and Transformer predictors. Positive values mean that shorter context increases error. The gray region marks the first five forecasts; bands show 95% confidence intervals over paired episode differences. (b) Transformer-source minus GRU-source history penalty over the first five forecasts and the full 44-step rollout. Positive values mean that the Transformer-trained representation depends more on long context.

Results Fig. 16 shows the results. (a) plots the change in error at each forecast step when the predictor receives four rather than 20 context frames. Negative values favor H4; positive values mean that shortening the context is harmful. We see that both representations benefit from H4 during the first two forecasts. From the third forecast onward, however, the Transformer-trained source remains above zero, while the GRU-trained source stays closer to zero and becomes consistently negative after $\mathfrak{h} = 22$. The GRU-trained representation therefore retains its rollout accuracy with substantially less

context. Panel (b) summarizes this divergence. The Transformer-source minus GRU-source history penalty is only 0.015 over the first five forecasts, but grows to 0.089 over the full 44-step rollout. The gap is therefore small at the start of prediction and becomes pronounced only after predicted latents are fed back recursively. This long-rollout robustness provides a concrete explanation for the stronger dynamics performance of the GRU-trained representation.

D.6 Performance on conventional control tasks

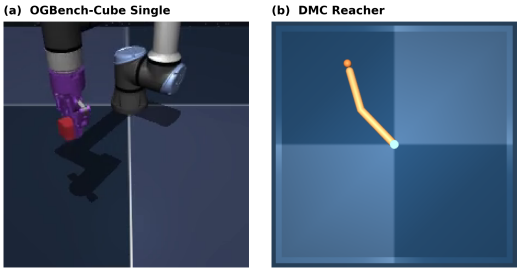


Figure 17: Representative illustration of the datasets (a) OGBench-Cube Single and (b) DMC Reacher.

The dynamics experiments above favor the GRU predictor over Transformer. We next test whether this choice degrades performance on conventional control tasks. We compare Transformer, GRU, and SSM predictors on OGBench-Cube Single and DMC Reacher (Figure 17), changing only the predictor architecture. All models use the same ViT-tiny encoder, H10/K5 objective, five-step action blocks and 20-epoch training schedule. We evaluate the same latent-space CEM planner on 50 fixed held-out cases per model.

Table 7: Predictor comparison on conventional control tasks. Success is measured over 50 fixed held-out cases, with episode-bootstrap 95% confidence intervals. Latent metrics use all 127,000 held-out H10/K5 clips.

Dataset	Predictor	Success ↑	K5 latent MSE ↓	Mean K1–K5 MSE ↓	Planning latency (s) ↓
OGBench-Cube	Transformer	68% (54–80)	0.00590	0.00445	0.195
	GRU	**70% (56–82)**	0.00561	0.00430	**0.111**
	SSM	68% (54–80)	**0.00497**	**0.00378**	0.279
DMC Reacher	Transformer	**100% (100–100)**	0.00226	0.00182	0.568
	GRU	98% (94–100)	0.00323	0.00289	**0.316**
	SSM	98% (94–100)	**0.00159**	**0.00132**	1.115

We observe that SG-JEPA (GRU/SSM) predictors preserves downstream control success on both tasks(Table 7). It also has the lowest planning latency and the highest effective rank on both datasets. Thus, the conventional benchmarks show no material loss in control success from using a GRU/SSM predictor. Together with its advantage on the dynamics benchmarks, this result supports our use of the GRU as the default predictor.

E 2D datasets full results

E.1 Gravity-conditioned rollouts

In Section 3 we present overall comparison on 2D triangle and square datasets. In this section we elaborate on the detailed breakdown, first by per-held-out-test-gravity evaluation metrics, as in Fig. 18; then results by rollout horizon, as in Table 8.

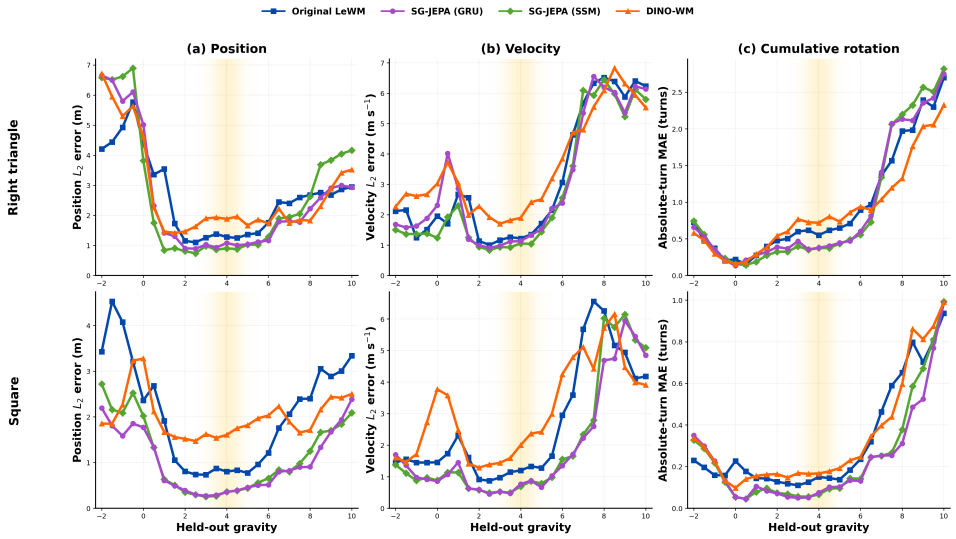


Figure 18: **Per-gravity evaluation at $\mathfrak{h}_{44}$.** Rows show right-triangle and square episodes; columns report planar position L_2 error, planar velocity L_2 error and cumulative-rotation MAE. The fading yellow background follows the relative training-gravity density $g \sim N(4, 0.5^2)$.

Metric	Method	$\mathfrak{h}_5$	$\mathfrak{h}_{16}$	$\mathfrak{h}_{32}$	$\mathfrak{h}_{44}$
Right triangle					
Position L_2 (m)	Original LeWM	1.105	1.943	**2.325**	2.625
	SG-JEPA (GRU)	0.968	2.167	2.486	**2.529**
	SG-JEPA (SSM)	1.251	2.455	2.628	2.665
	DINO-WM	**0.532**	**1.510**	2.329	2.764
Velocity L_2 (m/s)	Original LeWM	2.324	**1.598**	**2.499**	3.121
	SG-JEPA (GRU)	2.258	1.920	2.837	3.001
	SG-JEPA (SSM)	2.343	1.667	2.592	**2.765**
	DINO-WM	**1.897**	2.420	3.275	3.574
Cumulative rotation (turns)	Original LeWM	0.065	0.269	0.640	0.946
	SG-JEPA (GRU)	0.061	0.268	0.617	0.904
	SG-JEPA (SSM)	**0.059**	0.264	0.618	0.911
	DINO-WM	0.062	**0.255**	**0.609**	**0.875**
Square					
Position L_2 (m)	Original LeWM	0.930	1.934	2.042	2.074
	SG-JEPA (GRU)	**0.374**	**0.658**	**0.862**	**1.034**
	SG-JEPA (SSM)	0.395	0.950	1.094	1.148
	DINO-WM	0.454	1.117	1.570	2.006
Velocity L_2 (m/s)	Original LeWM	1.791	1.545	2.735	2.631
	SG-JEPA (GRU)	1.158	**0.896**	**1.455**	**1.915**
	SG-JEPA (SSM)	**1.051**	1.007	1.661	2.007
	DINO-WM	1.793	1.921	3.011	3.082
Cumulative rotation (turns)	Original LeWM	0.029	0.104	0.215	0.321
	SG-JEPA (GRU)	0.018	0.066	**0.143**	**0.236**
	SG-JEPA (SSM)	**0.015**	**0.061**	0.146	0.249
	DINO-WM	0.027	0.104	0.223	0.341

Table 8: **Rollout results**. Forecast steps $\mathfrak{h}_5, \mathfrak{h}_{16}, \mathfrak{h}_{32}$, and $\mathfrak{h}_{44}$ correspond to 0.3125, 1.0, 2.0, and 2.75 seconds. **bold** and underlining mark the best and second-best value within each shape, metric and step.

Results SG-JEPA gives the strongest overall performance, with its clearest advantage appearing as rollout errors accumulate. On the square, the GRU and SSM variants lead every metric at every reported step. SSM is slightly stronger at the shortest steps, while GRU is more accurate over longer recursive rollouts. Figure 18 shows that these gains extend across most of the held-out gravity grid, rather than being confined to the center of the training distribution. The few reversals occur mainly for velocity at the highest gravity values.

The triangle is more competitive at short and intermediate steps, where the baselines lead several translation entries. Their advantage does not persist: at $\mathfrak{h}_{44}$, SG-JEPA (GRU) has the lowest position error and SG-JEPA (SSM) has the lowest velocity error. DINO-WM retains the lowest cumulative-rotation error on the triangle, the only metric not led by SG-JEPA at the longest rollout. Across the two shapes, SG-JEPA therefore leads five of the six $\mathfrak{h}_{44}$ shape-metric comparisons, supporting its advantage for long-horizon dynamics prediction.

E.2 Shape generalization

In this separate ablation we study the model's ability of shape generalization. To do this we conduct the following experiment: We train SG-JEPA on both 2D right triangle and square datasets. And we ask whether it can generalize dynamics from these component shapes to the composite "house" shape (A "house" joins a right triangle and a square into one rigid body), which is unseen during training. We perform two tests, one trained with triangle + square ($T{+}S$), the other with triangle + square + pentagon ($T{+}S{+}P$). We then compare the performance against SG-JEPA trained with only "house" dataset on downstream probe evaluation. Moreover we trained with both Tiny ViT ($D = 192$) and Small ViT ($D = 384$) encoders for these cases to rule out the possibility of model capacity limit when dealing with multiple datasets. All six world models follow the planar SG-JEPA recipe in Appendix Section C.

Table 9: House rollout error at $\mathfrak{h}_{44}$, averaged over the seven gravity values in the band $g \in [2.5, 5.5]$.

Encoder	World-model training data	Position L_2 (m)	Velocity L_2 (m/s)	Cumulative rotation (turns)
Tiny	$T{+}S$	2.681	2.846	0.821
Tiny	$T{+}S{+}P$	2.733	2.792	0.832
Tiny	House-trained	**1.723**	**1.444**	**0.193**
Small	$T{+}S$	2.489	2.725	0.787
Small	$T{+}S{+}P$	2.556	2.762	0.833
Small	House-trained	**1.781**	**1.607**	**0.234**

Results In Table 9 and Fig. 19 we present the results. Overall, SG-JEPA transfers a substantial part of the house's translational dynamics from its component shapes. Position transfers most consistently, velocity is less stable, and rotation remains the hardest quantity to predict. Direct house training is still better near the training distribution. As shown in Table 9, within $g \in [2.5, 5.5]$, Small $T{+}S$ has $1.40\times$ the position error and $1.70\times$ the velocity error of the house-trained reference. Adding pentagons brings no consistent improvement, so the observed transfer already comes from the triangle and square components rather than from greater shape diversity.

The main highlight is the OOD gravity result in Figure 19. Across much of both gravity tails, the $T{+}S$ model closely follows the house-trained reference on position despite never seeing a house during world-model training. With the Small encoder, it matches or outperforms the house-trained model at several extreme gravity values.

The true-latent control helps localize the remaining gap. On Small $T{+}S$, probing encoded ground-truth house frames gives error ratios of 0.90 for position, 1.23 for velocity, and 1.12 for cumulative

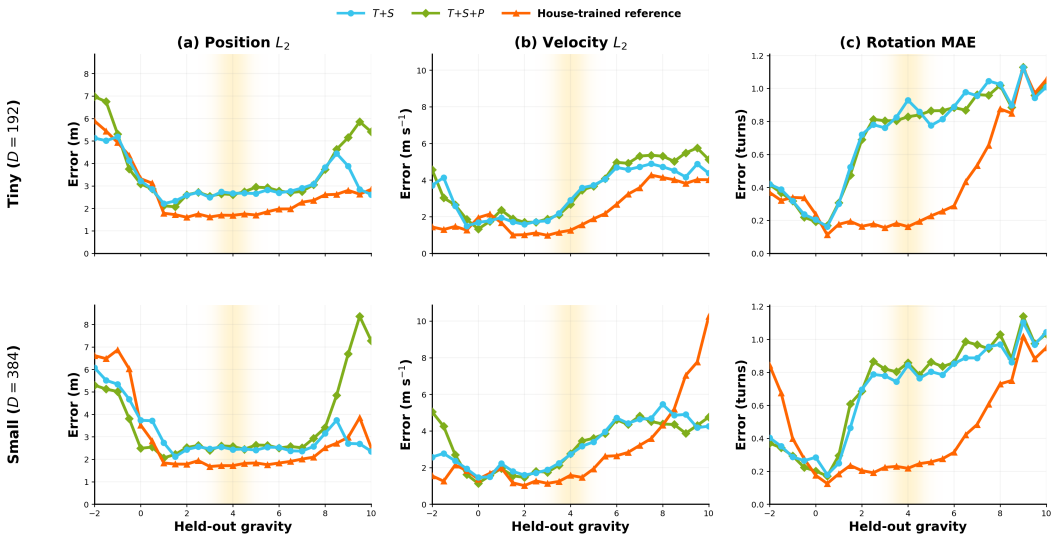


Figure 19: Per-gravity house rollout error at $\mathfrak{h}_{44}$. Rows show the Tiny ($D = 192$) and Small ($D = 384$) encoders; columns show position, velocity, and cumulative-rotation error.

rotation relative to the house-trained reference. After autoregressive rollout, the ratios rise to 1.40, 1.70, and 3.36. Thus most of the rotation gap develops during recursive prediction, where small angular-velocity errors accumulate and the new contact geometry matters.

F 3D DATASETS FULL RESULTS

F.1 APPROACH-BALL RESULTS

In main text, Fig. 4(a)(b) we report Approach-Ball position error over forecast steps and held-out gravity values. Here we show the complete results with velocity error and a compact summary of both physical quantities. Note that for this dataset we do not have ball rotation or friction, so the only physical quantities are position and velocity.

Table 10: Approach-Ball physical-state prediction error on 1,600 held-out episodes. The mean is taken over $\mathfrak{h}_1$–$\mathfrak{h}_{44}$, and $\mathfrak{h}_{44}$ is 2.75 s at 16 fps. Errors are reported in physical units. Lower is better; bold marks the best result.

	Position L_2 (m)		Velocity L_2 (m s^{-1})	
Method	Mean	$\mathfrak{h}_{44}$	Mean	$\mathfrak{h}_{44}$
Original LeWM	0.0986	0.1229	1.1361	1.1400
SG-JEPA (GRU)	**0.0491**	**0.0764**	**0.5391**	0.7829
SG-JEPA (SSM)	0.0495	0.0785	0.5664	0.8861
DINO-WM	0.0705	0.1152	0.5672	**0.7411**

Results Position gives the clearest result. The full state readout reinforces the finding in the main text: SG-JEPA (GRU) and SG-JEPA (SSM) have the two lowest position errors, both on average and at the final step, and reduce mean position error by roughly 30% relative to DINO-WM.

Velocity does not show the same uniform advantage however. GRU has a slightly lower rollout-average error than DINO-WM, while SSM and DINO-WM are nearly tied. DINO-WM is more accurate at $\mathfrak{h}_{44}$. As shown in Figure 20(a), GRU or SSM is lower over much of the intermediate rollout, but DINO-WM closes the gap and becomes stronger near the end.

The gravity breakdown in Figure 20(b) also separates the extrapolation regimes. GRU is strongest across most of the central and moderate-gravity range, DINO-WM at the high-gravity tail, and Original LeWM at the few lowest values. Taken together, these results show a clear SG-JEPA advantage for position. For velocity, SG-JEPA is competitive with, but not uniformly better than, DINO-WM. Each world model uses one training seed, so these episode averages do not measure variation across training runs.

F.2 ARM-CATCHER-BALL CONTROL RESULTS

In main text Figure 4(c,d) we present the success rate comparison for Arm Catcher Ball for SG-JEPA (GRU) and DINO-WM. In this section we report additional results for SG-JEPA (SSM) and Original LeWM (Transformer predictor) for completeness. Diffusion policies are trained for each checkpoint respectively. At inference, the policy conditions on a 20-frame latent history, predicts 16 Cartesian actions, executes eight, and replans. Sampling uses the complete 100-step cosine DDPM. Success requires the ball to enter the catcher and be latched before the episode ends. We evaluate the same test set at 23 gravity values under five paired simulator-rollout seeds.

Results In Fig. 21 we observe that SG-JEPA more than doubles the overall capture rate, improving on DINO-WM by 13.8%. The gain is consistent across all five paired rollouts, ranging from 12.9% to 14.3%. It is also not confined to the center of the training distribution. Fig. 4(d) shows a positive mean advantage at every gravity, with a significant gap from $g = 0$ through $g = 9$. At the training gravity, SG-JEPA reaches 42.3% compared with 16.0% for DINO-WM. Most remaining failures occur at the extreme gravity values or run to the full horizon without a capture. We also compare Original LeWM and both SG-JEPA predictors. Original LeWM uses its one-step Transformer objective

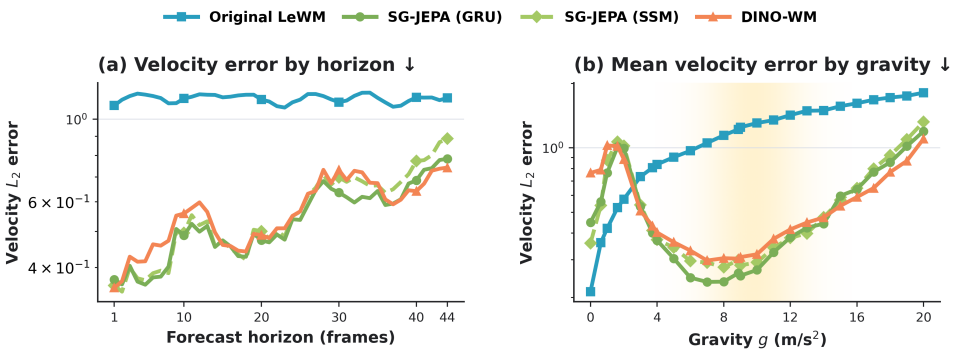


Figure 20: **Approach-Ball velocity prediction.** The evaluation and plotting setup matches the position results in Figure 4(a,b). (a) Mean ball-velocity L_2 error at each recursive forecast step. (b) Mean error over $\mathfrak{h}_1$–$\mathfrak{h}_{44}$ at each of the 25 held-out gravity values. The fading yellow background shows the training distribution $g \sim \mathcal{N}(9.8, 2.0^2)$.

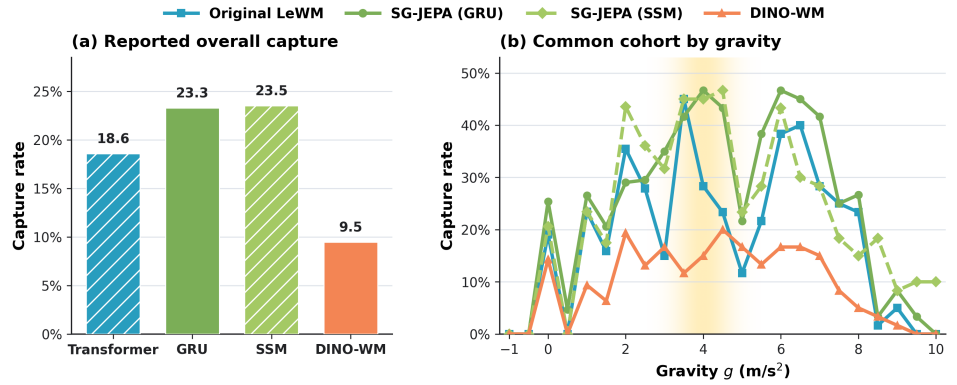


Figure 21: **Arm-Catcher-Ball predictor-family comparison.** (a) Overall capture rate. Results are averaged over 5 rollout executions. (b) Comparison of four methods by gravity. The fading yellow background shows the training distribution $g \sim \mathcal{N}(4, 0.5^2)$.

with $\lambda_{\text{SIG}} = 0.09$, while the GRU and SSM variants use the five-step SG-JEPA objective with $\lambda_{\text{SIG}} = 0.18$. Both SG-JEPA variants remain ahead of the baselines overall.

Overall, Arm Catcher Ball is a phase-sensitive interception task: successful control requires the policy to preserve the ball's velocity and contact phase, not merely its current position. As gravity increases, shorter bounce intervals amplify small timing errors and can shift the predicted trajectory to the wrong contact phase. SG-JEPA's consistent advantage across a broad gravity range therefore indicates that its representation retains more control-relevant dynamical information under changing physics. Performance nevertheless degrades under negative and extreme gravity values, where the motion departs qualitatively from the training regime.

F.3 Franka paddle-to-basket results

In main-text, Figu 4(c,e) compares SG-JEPA (GRU) and DINO-WM using five paired simulator-rollout seeds. Here we add results for Original LeWM and SG-JEPA (SSM) in Table 11 for completeness.

Results Both SG-JEPA pipelines give the highest strict-entry rates in the four-method comparison. DINO-WM hits the ball slightly more often and has the highest relaxed-success rate, but converts fewer hits into strict entries. The advantage of SG-JEPA therefore appears after contact, where the policy must produce the appropriate outgoing position and velocity rather than merely intercept the ball. GRU and SSM are effectively tied in this fixed rollout.

Table 11: **Four-method comparison.** Relaxed success records valid post-strike basket contact, while strict entry requires the ball to enter the basket. "Strict | blade" conditions strict entry on valid paddle-blade contact. All values are percentages.

Representation	Strict entry	Relaxed success	Blade contact	Strict | blade
SG-JEPA (GRU)	**30.36**	64.72	93.70	**32.40**
Original LeWM	28.80	62.00	90.46	31.84
SG-JEPA (SSM)	30.00	61.22	93.54	32.07
DINO-WM	27.60	**67.30**	**95.86**	28.79

In Fig. 22 we show the per-gravity breakdown for paddle hit and final basket entry success rates respectively. Paddle-hit rates remain high beyond the two weakest gravity values, while strict entry varies substantially, reinforcing that post-impact targeting is the harder stage. The entry ordering also changes sharply outside the center of the training distribution. The two largest reversals persist under all five rollout seeds: at $g = 13.9$ m/s^2, DINO-WM averages 61.2% strict entry and SG-JEPA (GRU) 1.8%; at $g = 16.0$ m/s^2, SG-JEPA averages 58.6% and DINO-WM 0.8%. The five runs use the same episode IDs, action scaling, and fixed checkpoint hashes, with the rollout seed as the only change. These are therefore reproducible task-specific reversals, although two gravity values are not enough to support a broader mechanistic conclusion.

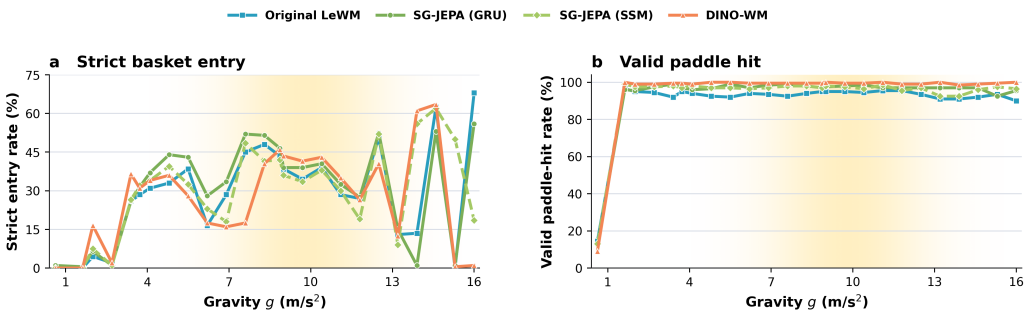


Figure 22: **Gravity-conditioned Franka Basket outcomes in the single-seed four-method comparison.** (a) Strict basket entry and (b) valid paddle-blade contact. Each point contains 200 paired episodes. The fading yellow background follows the training-gravity density $g \sim \mathcal{N}(9.8, 2.0^2)$. All control and evaluation settings are matched across methods.

The high paddle-hit rates show that interception is not the main bottleneck. Success instead depends on controlling the ball's post-impact state: small errors in contact timing, paddle orientation, or incoming velocity can change the outgoing impulse enough to miss the basket. Gravity further changes the required trajectory: weak gravity prolongs flight and can produce overshoot, whereas strong gravity demands a larger and more precisely directed impulse. SG-JEPA's higher entry rate therefore suggests that its representation better preserves the velocity and contact-phase information needed for gravity-dependent impact control.

F.4 Arm-Paddle-Ball results

Arm Paddle Ball requires repeated contact control: the policy must keep the ball off the floor and produce at least one qualified bounce. Main-text Fig. 4(c,f) compares the final policy configurations. Here we evaluate the model's capability of understanding the ball dynamics by training probes and evaluated on held-out test set.

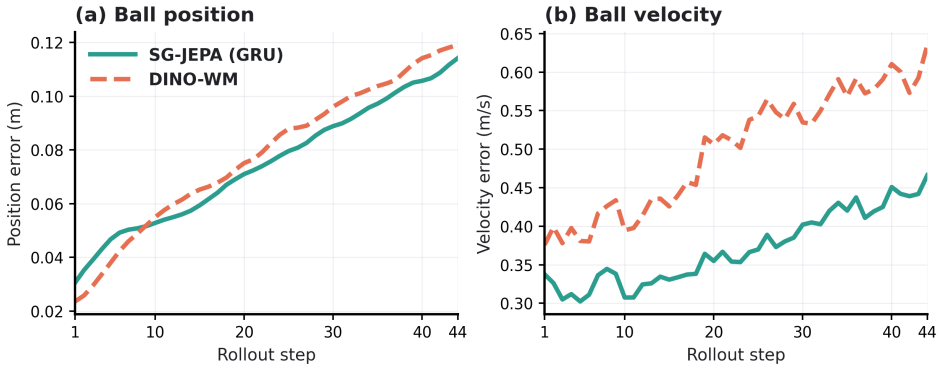


Figure 23: **Open-loop Arm-Paddle-Ball prediction.** Mean Euclidean error for (a) ball position and (b) ball velocity over 5,000 held-out episodes. SG-JEPA has lower position error from $\mathfrak{h}_9$ onward and lower velocity error at every forecast step.

Prediction results. SG-JEPA has lower position error at 36 of 44 forecast steps and lower velocity error at all 44. At $\mathfrak{h}_{44}$, position error falls from 0.119 to 0.114 m and velocity error from 0.635 to 0.466 m/s. The larger velocity gap is relevant to sustained bouncing, where small phase errors carry into the next contact. This probe comparison is evidence for more accurate open-loop dynamics.

G Sparse-Gravity Post-Training Ablation

In this ablation we study whether post-training on a subset of the test set with out-of-distribution gravity values improves long-horizon dynamics prediction between an beyond these values. We evaluate 2D right triangle and square datasets and four methods (Original LeWM, SG-JEPA (GRU), SG-JEPA (SSM) and DINO-WM).

The target post-training set contains 3,200 episodes: 800 episodes at each of the four support gravity values $\mathcal{G}_{\text{sup}} = \{0, 2, 6, 8\}$. The *target-only* arm uses these 3,200 episodes. The *mixed* arm adds 5,000 episodes sampled from the original source-training distribution, for 8,200 episodes in total. So for each test value it's roughly $85 : 15\%$ split. These data are excluded from later evaluation to avoid leakage. Each model is warm-started from its corresponding pretrained source checkpoint and optimized for 20 epochs with a fresh optimizer. We then report rollout errors at horizon 44. The held-out evaluation partition used here contains 2,500 episodes, with 100 episodes at every gravity in $\{-2, -1.5, \ldots, 9.5, 10\}$. For condition c, metric m, and gravity g, let $E_{c,m}(g)$ be the mean error and let $E_{0,m}(g)$ denote the corresponding pretrained error. The aggregate improvement is

$$\Delta_c = 100 \left[1 - \frac{1}{3|\mathcal{G}_{\text{int}}|} \sum_{m=1}^{3} \sum_{g \in \mathcal{G}_{\text{int}}} \frac{E_{c,m}(g)}{E_{0,m}(g)} \right], \tag{8}$$

where $\mathcal{G}_{\text{int}} = \{0, 0.5, \ldots, 8\} \setminus \mathcal{G}_{\text{sup}}$ contains the 13 held-out interpolation gravity values. Thus, a positive value indicates a reduction in error relative to the same model's pretrained baseline. We also check whether post-training preserves performance near the source distribution. A model passes this *source-range retention* test if, for each metric, its mean error over $g \in \{2.5, 3, \ldots, 5.5\}$ is no more than 5% above its pretrained error.

Results Table 12 and Figures 24–25 support four main observations.

- Mixed post-training reduces aggregate error relative to the pretrained model in all eight shape and model combinations. It also outperforms target-only post-training in every combination. The mean reduction increases from 6.85% to 14.05%, a gain of 7.21 percentage points.
- SG-JEPA (GRU) improves by 20.84% on the triangle and 13.60% on the square. The largest mixed improvement for each shape is 20.84% for triangle SG-JEPA (GRU) and 20.81% for square Original LeWM.

- The per-gravity curves show that adaptation does more than improve the four observed support gravity values. Both post-training conditions reduce error at many of the unobserved gravity values between them, and the largest separations from the pretrained curves generally occur toward the low- and high-gravity ends. The effect is therefore consistent with interpolation across the adapted gravity range.
- The gains are not uniform across all metrics. Target-only DINO-WM slightly worsens triangle rotation by 0.22%, while Mixed DINO-WM worsens square velocity by 4.13%. Around the central source range, where the pretrained curves are already lower, the changes are smaller and more method dependent. Four of the eight Mixed configurations pass the source-range retention test; none of the Target-only configurations pass it.

The consistent advantage of Mixed over Target-only suggests that a modest amount of data at a few target gravity values can improve prediction at the unobserved gravity values between them. Source replay also gives stronger aggregate performance than using the target trajectories alone. In settings where the governing dynamics change, this offers an alternative to training a new world model from scratch: collect a small target set and post-train with source replay.

Shape	Predictor	Arm	Overall	Position	Velocity	Rotation	Source retained
Triangle	Transformer	Target-only	4.22	5.30	5.65	1.70	no
	Transformer	Mixed	13.80	17.29	13.64	10.47	no
	GRU	Target-only	17.05	17.04	20.83	13.29	no
	GRU	Mixed	20.84	19.69	21.63	21.20	yes
	SSM	Target-only	6.61	2.73	11.29	5.80	no
	SSM	Mixed	14.53	12.93	15.93	14.74	yes
	DINO-WM	Target-only	1.79	2.92	2.66	−0.22	no
	DINO-WM	Mixed	6.12	5.38	7.10	5.89	yes
Square	Transformer	Target-only	10.41	11.46	8.21	11.58	no
	Transformer	Mixed	20.81	25.43	18.48	18.51	no
	GRU	Target-only	5.39	7.99	6.61	1.56	no
	GRU	Mixed	13.60	9.09	9.60	22.12	yes
	SSM	Target-only	3.71	6.96	2.51	1.66	no
	SSM	Mixed	12.15	17.84	10.95	7.66	no
	DINO-WM	Target-only	5.59	15.85	0.05	0.86	no
	DINO-WM	Mixed	10.55	21.43	−4.13	14.35	no
Target-only mean			6.85	8.78	7.22	4.53	—
Mixed mean			14.05	16.14	11.65	14.37	—

Table 12: Improvement (%) on the 13 held-out interpolation gravity values, relative to each predictor's pretrained baseline. "Overall" is Δ_c from Eq. (8); "Source retained" reports the source-range retention test defined above.

Figures 24 and 25 provide the complete per-gravity breakdown.



Figure 24: Right-triangle horizon-44 errors. Rows correspond to the four predictors and columns to position, velocity, and cumulative rotation. Curves show the pretrained baseline, target-only post-training, and mixed post-training over all 25 evaluation gravity values.



Figure 25: Square horizon-44 errors at epoch 20, using the same layout and conventions as Fig. 24.

H THEORY OF PREDICTION UNDER CHANGING PHYSICAL LAWS

This appendix develops the theory used in Section 4. The analysis separates three questions. First, what makes a latent state locally predictive when gravity changes? Second, how does a local error enter a recursive rollout? Third, why can a multi-step objective prefer a different representation from a one-step objective, e.g. one-step favours DINO-WM but after rollout SG-JEPA has advantage? We answer these questions with a linear feature model. The model is simple enough to analyze exactly, but it is not an identification of the neural encoder or the GRU with linear operators.

We first summarize the main results.

- The one-step latent error separates into predictor error on learned features, information discarded by the representation that still affects the next state, and a conditionally zero-mean residual (Definition H.1 and Proposition H.2).
- If the true and learned dynamics share the same dependence on gravity, average training error bounds the systematic one-step test error, scaled by gravity coverage (Theorem H.4). For affine free flight, the factor is $1 + (g_\star - \mu_{\mathrm{tr}})^2/\sigma_{\mathrm{tr}}^2$.
- Rollout recursion shows how local error and transition residual are transformed by later predicted dynamics. The corresponding physical-state bound also retains readout error and the transition residual (Theorems H.10 and H.12).
- A multi-step loss applies a different quadratic geometry to one-step transition error. Two representations can therefore exchange rank when the training horizon changes (Theorem H.15).

H.1 SETUP: THE MODEL

The model mirrors the training objective and fixes the notation used in the proofs.

Implementation-aligned objective. Let e_ϑ be the shared trainable encoder and projector, and denote the latent vector $z_t = e_\vartheta(o_t) \in \mathbb{R}^{d_z}$ for observation o_t. Let $c_t = q(u_t, g)$ be the action-conditioning of the action u_t and gravity g. The predictor receives a context of $H \geq 1$ latent states. Starting from a true context, it is applied recursively for $K \geq 1$ training steps:

$$\widehat{z}_{t+k} = f\left(z^{\text{roll}}_{t+k-H:t+k-1}, c_{t+k-H:t+k-1}\right), \qquad k = 1, \ldots, K, \tag{9}$$

$$z^{\text{roll}}_{t+j} = \begin{cases} z_{t+j}, & j \leq 0, \\ \widehat{z}_{t+j}, & j \geq 1. \end{cases} \tag{10}$$

For normalized weights

$$w_k = \frac{\gamma^{k-1}}{\sum_{j=1}^{K} \gamma^{j-1}}, \qquad \gamma > 0, \qquad w_k \geq 0, \qquad \sum_{k=1}^{K} w_k = 1, \tag{11}$$

the latent rollout loss is

$$\mathcal{L}_K(\vartheta, f) = \sum_{k=1}^{K} w_k \, \mathbb{E}\left[\|\widehat{z}_{t+k} - z_{t+k}\|_2^2\right]. \tag{12}$$

The expectation is over the training clips, including initial states, gravity values, action sequences, and stochastic residuals when present. The predictor helps determine which representations receive low training loss. We express this dependence through the profiled representation risk

$$\mathcal{J}_{K,H,\mathcal{F}}(\vartheta) = \inf_{f \in \mathcal{F}_H} \mathcal{L}_K(\vartheta, f) + \lambda_{\text{SIG}} \mathcal{R}_{\text{SIG}}(\vartheta). \tag{13}$$

Here $\mathcal{F}_H$ is the predictor family for an H-step context. Unless marked as empirical, $\mathcal{J}_{K,H,\mathcal{F}}$ denotes a population risk; $\mathcal{R}_{\text{SIG}}$ is the expected implemented batch statistic, with the expectation taken over data batches and random directions. The idealized population characteristic-function calculation is defined separately in Section H.8.

For the core argument, let scalar gravity $g \in \mathbb{R}$ be fixed within an episode. We assume that the feature state $\phi_t \in \mathbb{R}^{d_\phi}$ follows the linear transition

$$\phi_{t+1} = T(g)\phi_t + \xi_{t+1} \qquad \mathbb{E}[\xi_{t+1} \mid \phi_t, g] = 0. \tag{14}$$

When actions are present explicitly, the conditioning set in this equation also includes the known action input. The matrix $T(g) \in \mathbb{R}^{d_\phi \times d_\phi}$ gives the transition under gravity g, and ξ_{t+1} is the remaining conditionally zero-mean residual. Systematic approximation error does not belong in ξ_{t+1}; it must enter a separate deterministic term, such as the approximate-basis remainder introduced below. Section H.6 gives the exact finite-history form with actions. We model the learned representation as

$$z_t = W\phi_t, \qquad W \in \mathbb{R}^{d_z \times d_\phi}, \qquad d_z \leq d_\phi, \qquad WW^\top = I_{d_z}. \tag{15}$$

Row orthonormality fixes the latent coordinate scale inside the model. The learned one-step predictor is represented by $\widehat{A}(g) \in \mathbb{R}^{d_z \times d_z}$. For the physical-state bounds, we assume that the physical quantity of interest is linear in the chosen feature coordinates:

$$s_t = C_{\text{phys}}\phi_t, \tag{16}$$

where $C_{\text{phys}} \in \mathbb{R}^{d_s \times d_\phi}$. Let $r : \mathbb{R}^{d_z} \to \mathbb{R}^{d_s}$ be a frozen probe. When the probe is linear, we write $r(z) = Rz$ with $R \in \mathbb{R}^{d_s \times d_z}$. The experiments use learned probes, some of which read a short latent history. For a history probe, the input to r is the corresponding stacked latent window. The exact selector for a linear history readout is given in Section H.6. The probe identities below hold for any frozen probe with a compatible input, while the matrix bounds use the linear specialization. s_t is expressed in the fixed physical coordinates in which error is evaluated. We use H only for context length, K only for the training rollout length, and $\mathfrak{h}$ for a generic evaluation horizon. Vector norms are Euclidean. Matrix norms are Frobenius norms unless a subscript 2 marks the spectral norm.

Fixed-gravity composition. For a fixed gravity and sampling interval, define

$$S_g(0) = I_{d_\phi}, \qquad S_g(\mathfrak{h}) = T(g)^{\mathfrak{h}}, \ \widehat{S}_g(0) \qquad = I_{d_z}, \widehat{S}_g(\mathfrak{h}) \qquad = \widehat{A}(g)^{\mathfrak{h}}. \tag{17}$$

For $k, \ell \in \mathbb{N}_0$,

$$S_g(k+\ell) = S_g(\ell)S_g(k), \qquad \widehat{S}_g(k+\ell) = \widehat{S}_g(\ell)\widehat{S}_g(k). \tag{18}$$

Each gravity therefore indexes a pair of discrete evolution semigroups; gravity is an index, not the composition variable. In the residual-free model these operators describe the realized evolution. With conditionally zero-mean residuals, the true powers describe conditional-mean propagation, while realized trajectories retain the residual terms derived below. The rollout loss trains the accuracy of the first K learned iterates rather than the algebraic composition identity itself.

H.2 Local prediction error under a fixed gravity

A latent state can fail locally for two reasons. It may discard information needed for the next transition, or the predictor may evolve the retained information incorrectly. The operators below separate these terms. Define the orthogonal projector onto the row space of W and the two induced operators

$$P_W = W^\top W, \quad A_W(g) = WT(g)W^\top, \quad C_W(g) = WT(g)(I - P_W). \tag{19}$$

Since $P_W\phi_t = W^\top z_t$,

$$WT(g)\phi_t = WT(g)P_W\phi_t + WT(g)(I - P_W)\phi_t \tag{20}$$

$$= A_W(g)z_t + C_W(g)\phi_t. \tag{21}$$

Equivalently, $C_W(g) = WT(g) - A_W(g)W$. Encoding Eq. 14 therefore gives

$$z_{t+1} = A_W(g)z_t + C_W(g)\phi_t + W\xi_{t+1}. \tag{22}$$

Definition H.1 (Local law-conditioned error)**.** Let $E_A(g) = A_W(g) - \widehat{A}(g)$. The state-dependent conditional-mean error injected when the predictor starts from the true latent state is

$$\delta_t(g) = E_A(g)z_t + C_W(g)\phi_t. \tag{23}$$

The term $E_A(g)z_t$ is predictor error on the state. The term $C_W(g)\phi_t$ is the closure error realized at state ϕ_t: discarded features still affect the next latent. The operator $C_W(g)$ can be nonzero even on trajectories for which this realized vector happens to vanish.

We call the representation *predictively closed* at gravity g when $C_W(g) = 0$. In that case, the conditional mean of the next latent depends only on the current latent. Residual variation can still make individual transitions unpredictable.

In the residual-free linear model, predictive closure together with an exact latent operator, $\widehat{A}(g) = A_W(g)$, gives $WT(g) = \widehat{A}(g)W$. Hence

$$WS_g(\mathfrak{h}) = \widehat{S}_g(\mathfrak{h})W \qquad \text{for every } \mathfrak{h} \geq 0. \tag{24}$$

With transition residuals, this intertwining identity applies to the conditional-mean evolution; individual trajectories still contain the propagated $W\xi_{t+1}$ terms.

Proposition H.2 (What teacher forcing measures)**.** *Let* $\widehat{z}^{\mathrm{TF}}_{t+1} = \widehat{A}(g)z_t$. *Then*

$$z_{t+1} - \widehat{z}^{\mathrm{TF}}_{t+1} = \delta_t(g) + W\xi_{t+1}, \tag{25}$$

$$Rz_{t+1} - R\widehat{z}^{\mathrm{TF}}_{t+1} = R\delta_t(g) + RW\xi_{t+1}, \tag{26}$$

$$s_{t+1} - R\widehat{z}^{\mathrm{TF}}_{t+1} = (C_{\mathrm{phys}} - RW)\phi_{t+1} + R\delta_t(g) + RW\xi_{t+1}. \tag{27}$$

Proof. Subtract $\widehat{A}(g)z_t$ from Eq. 22. This gives Eq. 25; applying R gives Eq. 26. For the physical identity, add and subtract $RW\phi_{t+1} = Rz_{t+1}$:

$$s_{t+1} - R\widehat{z}^{\mathrm{TF}}_{t+1} = C_{\mathrm{phys}}\phi_{t+1} - RW\phi_{t+1} + R(z_{t+1} - \widehat{z}^{\mathrm{TF}}_{t+1}), \tag{28}$$

and substitute Eq. 25. □

For any frozen, possibly nonlinear probe r, the decoded-latent comparison is $r(z_{t+1}) - r(\widehat{z}^{\mathrm{TF}}_{t+1})$. The readout residual $s_{t+1} - r(z_{t+1})$ cancels because the same probe is applied to both latents. If r is L_r-Lipschitz on the line segment joining the two latents, then

$$\left\| r(z_{t+1}) - r(\widehat{z}^{\mathrm{TF}}_{t+1}) \right\|_2 \le L_r \left\| \delta_t(g) + W\xi_{t+1} \right\|_2 . \tag{29}$$

The teacher-forced diagnostic in Figure 5 uses this same-probe comparison, so the true-state probe floor cancels for both linear and nonlinear probes. The remaining error still combines the two terms in $\delta_t(g)$ with the transition residual. Under the conditional-zero-mean assumption, the expected cross term also vanishes after any fixed linear probe:

$$\mathbb{E}\big[\delta_t(g)^\top R^\top R W \xi_{t+1} \mid g\big] = 0. \tag{30}$$

This follows from the tower property because $\delta_t(g)$ is measurable with respect to (ϕ_t, g) and $\mathbb{E}[\xi_{t+1} \mid \phi_t, g] = 0$. There is no corresponding cancellation guarantee for a nonlinear probe, and the result says nothing about the readout-residual cross term in Eq. 73. The diagnostic cannot, by itself, assign the difference to $E_A(g)z_t$ or $C_W(g)\phi_t$ separately.

H.3 How gravity coverage controls test error

The semigroup law composes evolution over $\mathfrak{h}$ at fixed g. Generalization across gravity is a separate question: it depends on how the one-step operator varies with g and which law directions training covers. Low error on the training gravity values alone gives no bound at an unseen gravity, so we state the required shared structure explicitly.

Shared law basis Suppose a column vector $\psi(g) = (\psi_1(g), \dots, \psi_m(g))^\top$ describes how gravity enters the transition, with finite second moments under the training distribution, and

$$T(g) = \sum_{k=1}^{m} \psi_k(g) T_k, \quad A_W(g) = \sum_{k=1}^{m} \psi_k(g) A_k, \tag{31}$$

$$C_W(g) = \sum_{k=1}^{m} \psi_k(g) C_k, \quad \widehat{A}(g) = \sum_{k=1}^{m} \psi_k(g) \widehat{A}_k. \tag{32}$$

The first expansion implies the next two with $A_k = WT_kW^\top$ and $C_k = WT_k(I - P_W)$. The expansion for $\widehat{A}(g)$ is an additional structural assumption on the learned predictor. Equation equation 32 is therefore an exact-model assumption for the result below, not a consequence of providing g as an input. The neural predictor is not constrained to satisfy it exactly.

Let P_{tr} be the training distribution over gravity and define

$$M_\psi = \mathbb{E}_{g\sim P_{\mathrm{tr}}}[\psi(g)\psi(g)^\top]. \tag{33}$$

We assume $M_\psi \succ 0$, so the training distribution covers every direction in the chosen basis. The *law-coverage factor* at a test gravity $g_\star$ is

$$\mathcal{L}_{\mathrm{law}}(g_\star) = \psi(g_\star)^\top M_\psi^{-1} \psi(g_\star). \tag{34}$$

A large value means that the test law lies in a weakly covered direction of the training design.

Lemma H.3 (Transfer for a matrix-valued law family)**.** *Let $E(g) = \sum_{k=1}^m \psi_k(g) E_k$ be any matrix-valued residual and set $\epsilon_E^2 = \mathbb{E}_{g\sim P_{\mathrm{tr}}}\left[\|E(g)\|_F^2\right]$. If $M_\psi \succ 0$, then*

$$\|E(g_\star)\|_F^2 \le \mathcal{L}_{\mathrm{law}}(g_\star)\epsilon_E^2. \tag{35}$$

Proof. Let E be the matrix whose kth column is $\mathrm{vec}(E_k)$. Then $\mathrm{vec}(E(g)) = \mathsf{E}\psi(g)$ and

$$\epsilon_E^2 = \mathbb{E}\big[\psi(g)^\top \mathsf{E}^\top \mathsf{E} \psi(g)\big] \tag{36}$$

$$= \mathrm{tr}(\mathsf{E} M_\psi \mathsf{E}^\top) = \|\mathsf{E} M_\psi^{1/2}\|_F^2. \tag{37}$$

At $g_\star$,

$$\|E(g_\star)\|_F = \left\| \mathsf{E} M_\psi^{1/2} M_\psi^{-1/2} \psi(g_\star) \right\|_2 \tag{38}$$

$$\leq \|\mathsf{E} M_\psi^{1/2}\|_2 \left\| M_\psi^{-1/2} \psi(g_\star) \right\|_2 \tag{39}$$

$$\leq \|\mathsf{E} M_\psi^{1/2}\|_F \sqrt{\mathcal{L}_{\text{law}}(g_\star)}. \tag{40}$$

Substitute Eq. 37 and square both sides. □

Define the average training-law errors

$$\epsilon_{\text{cl}}^2 = \mathbb{E}_{g\sim P_{\text{tr}}} \left[\|C_W(g)\|_F^2 \right], \tag{41}$$

$$\epsilon_{\text{op}}^2 = \mathbb{E}_{g\sim P_{\text{tr}}} \left[\|A_W(g) - \widehat{A}(g)\|_F^2 \right]. \tag{42}$$

The first measures how far the representation is from predictive closure. The second measures how well the predictor fits the transition retained by the representation.

Theorem H.4 (Local error at an unseen gravity)**.** *Under the shared-basis assumptions in Eq. 32, suppose* $\|z_t\|_2 \leq B_z$ *and* $\|\phi_t\|_2 \leq B_\phi$*. Then*

$$\|\delta_t(g_\star)\|_2 \leq \sqrt{\mathcal{L}_{\text{law}}(g_\star)} \left(B_z \epsilon_{\text{op}} + B_\phi \epsilon_{\text{cl}} \right). \tag{43}$$

Proof. Apply Lemma H.3 first to $C_W(g)$ and then to $E_A(g) = A_W(g) - \widehat{A}(g)$. This gives

$$\|C_W(g_\star)\|_F \leq \sqrt{\mathcal{L}_{\text{law}}(g_\star)} \epsilon_{\text{cl}}, \tag{44}$$

$$\|E_A(g_\star)\|_F \leq \sqrt{\mathcal{L}_{\text{law}}(g_\star)} \epsilon_{\text{op}}. \tag{45}$$

Using Definition H.1, the triangle inequality, and $\|Mx\|_2 \leq \|M\|_F \|x\|_2$ yields

$$\|\delta_t(g_\star)\|_2 \leq \|E_A(g_\star)\|_F \|z_t\|_2 + \|C_W(g_\star)\|_F \|\phi_t\|_2, \tag{46}$$

which gives the stated bound. □

The theorem is a sufficient transfer result. As a concrete example, for collision-free ballistic motion, gravity enters the transition affinely. This gives a concrete two-dimensional law basis.

Corollary H.5 (Coverage factor for affine gravity)**.** *Let* $\psi(g) = (1, g)^\top$*. If the training gravity has mean* μ_{tr} *and variance* $\sigma_{\text{tr}}^2 > 0$*, then*

$$\mathcal{L}_{\text{law}}(g_\star) = 1 + \frac{(g_\star - \mu_{\text{tr}})^2}{\sigma_{\text{tr}}^2}. \tag{47}$$

Proof. For this basis,

$$M_\psi = \begin{bmatrix} 1 & \mu_{\text{tr}} \\ \mu_{\text{tr}} & \mu_{\text{tr}}^2 + \sigma_{\text{tr}}^2 \end{bmatrix}. \tag{48}$$

Its determinant is σ_{tr}^2, so

$$M_\psi^{-1} = \frac{1}{\sigma_{\text{tr}}^2} \begin{bmatrix} \mu_{\text{tr}}^2 + \sigma_{\text{tr}}^2 & -\mu_{\text{tr}} \\ -\mu_{\text{tr}} & 1 \end{bmatrix}. \tag{49}$$

Direct multiplication gives

$$\begin{bmatrix} 1 & g_\star \end{bmatrix} M_\psi^{-1} \begin{bmatrix} 1 \\ g_\star \end{bmatrix} = \frac{\mu_{\text{tr}}^2 + \sigma_{\text{tr}}^2 - 2\mu_{\text{tr}} g_\star + g_\star^2}{\sigma_{\text{tr}}^2} \tag{50}$$

$$= 1 + \frac{(g_\star - \mu_{\text{tr}})^2}{\sigma_{\text{tr}}^2}. \tag{51}$$

□

The moments in Eq. 47 are the mean and variance of the actual training distribution. The planar experiments draw $g = \max\{G, 0.1\}$ with $G \sim \mathcal{N}(4, 0.5^2)$, so the exact factor uses the moments of this clipped distribution.

Proposition H.6 (Free-flight dynamics remain affine in gravity)**.** *For time step $\Delta > 0$ and homogeneous state $\phi_t = (p_t, v_t, 1)^\top$, let*

$$T_g = \begin{bmatrix} 1 & \Delta & -\frac{1}{2}g\Delta^2 \\ 0 & 1 & -g\Delta \\ 0 & 0 & 1 \end{bmatrix}. \tag{52}$$

For every integer $\mathfrak{h} \geq 1$,

$$T_g^{\mathfrak{h}} = \begin{bmatrix} 1 & \mathfrak{h}\Delta & -\frac{1}{2}g(\mathfrak{h}\Delta)^2 \\ 0 & 1 & -g\mathfrak{h}\Delta \\ 0 & 0 & 1 \end{bmatrix}. \tag{53}$$

Thus the transition remains affine in g at every free-flight horizon.

Proof. The formula holds at $\mathfrak{h} = 1$. Assume it holds at $\mathfrak{h}$. Multiplying $T_g^{\mathfrak{h}}$ by T_g gives the next velocity coefficient $-g(\mathfrak{h}+1)\Delta$. The homogeneous contribution to position is

$$-\tfrac{1}{2}g(\mathfrak{h}\Delta)^2 - \mathfrak{h}g\Delta^2 - \tfrac{1}{2}g\Delta^2 = -\tfrac{1}{2}g(\mathfrak{h}+1)^2\Delta^2. \tag{54}$$

The remaining entries also match Eq. 53 with $\mathfrak{h}$ replaced by $\mathfrak{h}+1$. Induction completes the proof. □

This proposition applies only while the trajectory remains in free flight. It is a statement about the displayed physical-coordinate feature state; an arbitrary learned feature map need not inherit the same affine dependence. In more than one spatial dimension, the same calculation applies to the gravity-aligned position and velocity components, while tangential components have no gravity forcing in this idealized free-flight model. Contact can change the transition branch, as discussed in Section H.7.

H.3.1 When the coverage result does not apply

The coverage theorem needs both a covered basis and a predictor that uses that basis.

Proposition H.7 (Missing law directions prevent a uniform bound)**.** *Suppose M_ψ is singular and there is an $a \in \ker(M_\psi)$ such that $a^\top \psi(g_\star) \neq 0$. Then a residual family can have zero average training error and nonzero error at $g_\star$.*

Proof. Because $a^\top M_\psi a = \mathbb{E}[(a^\top \psi(g))^2] = 0$, we have $a^\top \psi(g) = 0$ almost surely under P_{tr}. Choose any nonzero matrix E_0 and set $E(g) = (a^\top \psi(g))E_0$. This residual vanishes almost surely on the training distribution, but it is nonzero at $g_\star$. □

A singular M_ψ is not automatically fatal. If $\psi(g_\star)$ lies in the covered range of M_ψ, the same type of statement can use the inverse restricted to that range. The failure above occurs when the test law has a component in a direction that training never observes.

Corollary H.8 (Transfer from a finite gravity design)**.** *Let $g_1, \ldots, g_n$ be observed gravity values and let $\Psi \in \mathbb{R}^{n\times m}$ have rows $\psi(g_i)^\top$. For a residual $E(g) = \sum_k \psi_k(g)E_k$, suppose Ψ has full column rank. Then*

$$\|E(g_\star)\|_F^2 \leq \psi(g_\star)^\top (\Psi^\top \Psi)^{-1} \psi(g_\star) \sum_{i=1}^{n} \|E(g_i)\|_F^2. \tag{55}$$

Proof. Let $Y \in \mathbb{R}^{n\times q}$ have row $\mathrm{vec}(E(g_i))^\top$, where q is the number of entries in each residual matrix. If $D \in \mathbb{R}^{m\times q}$ stacks the vectorized coefficient matrices, then $Y = \Psi D$. Full column rank gives $D = (\Psi^\top\Psi)^{-1}\Psi^\top Y$. Hence

$$\mathrm{vec}(E(g_\star))^\top = \psi(g_\star)^\top(\Psi^\top\Psi)^{-1}\Psi^\top Y. \tag{56}$$

Cauchy-Schwarz bounds its squared norm by the squared norm of the row vector times $\|Y\|_F^2$. The row-vector norm is $\psi(g_\star)^\top(\Psi^\top\Psi)^{-1}\psi(g_\star)$, which proves the claim. □

Theorem H.9 (Conditioning alone does not guarantee extrapolation)**.** *Let $g_1, \dots, g_n$ be distinct observed gravity values and let $g_\star$ be different from all of them. For any analytic matrix-valued function $\widehat{A}_0(g)$ and any matrix Q, there is an analytic matrix-valued function $\widehat{A}_1(g)$ such that*

$$\widehat{A}_1(g_i) = \widehat{A}_0(g_i) \quad \textit{for every } i, \qquad \widehat{A}_1(g_\star) = \widehat{A}_0(g_\star) + Q. \tag{57}$$

Proof. Define

$$q(g) = \frac{\prod_{i=1}^n (g - g_i)}{\prod_{i=1}^n (g_\star - g_i)}. \tag{58}$$

The denominator is nonzero, $q(g_i) = 0$, and $q(g_\star) = 1$. The function $\widehat{A}_1(g) = \widehat{A}_0(g) + q(g)Q$ is analytic and has the required values. □

Even an analytic conditional model can agree at every observed gravity and differ arbitrarily at an unseen one. Supplying gravity is therefore not, by itself, an extrapolation guarantee.

If a residual has the approximate form

$$E(g) = \sum_{k=1}^m \psi_k(g)E_k + \rho_E(g), \tag{59}$$

write

$$E_{\mathrm{basis}}(g) = \sum_{k=1}^m \psi_k(g)E_k, \qquad \epsilon^2_{E,\mathrm{basis}} = \mathbb{E}_{g\sim P_{\mathrm{tr}}}\|E_{\mathrm{basis}}(g)\|_F^2. \tag{60}$$

Lemma H.3 applies to E_{basis} and gives

$$\|E(g_\star)\|_F \le \sqrt{\mathcal{L}_{\mathrm{law}}(g_\star)}\epsilon_{E,\mathrm{basis}} + \|\rho_E(g_\star)\|_F. \tag{61}$$

Here $\epsilon_{E,\mathrm{basis}}$ is not automatically bounded by the observed training residual: the basis component and ρ_E may cancel at the training gravity values. For E_A, call the remainder ρ_{op}; for C_W, call it ρ_{cl}. Applying the formula to both residuals adds $B_z\|\rho_{\mathrm{op}}(g_\star)\|_F + B_\phi\|\rho_{\mathrm{cl}}(g_\star)\|_F$ to the right-hand side of Eq. 43. Training error cannot control a remainder that appears only in a region with little or no training mass.

H.4 Local errors under repeated composition

Teacher forcing measures the local error before predictions are reused. Free rollout feeds each predicted latent back into the predictor, so later learned iterates act on every earlier defect. The following recursion makes this dependence exact.

At a fixed gravity g, let the true and predicted latent transitions be

$$z_{j+1} = A_W(g)z_j + C_W(g)\phi_j + W\xi_{j+1}, \tag{62}$$

$$\widehat{z}_{j+1} = \widehat{A}(g)\widehat{z}_j, \qquad \widehat{z}_0 = z_0. \tag{63}$$

Define the rollout error $e_j = z_j - \widehat{z}_j$.

Theorem H.10 (Exact recursive latent error)**.** *For every $j \ge 0$,*

$$e_{j+1} = \widehat{A}(g)e_j + \delta_j(g) + W\xi_{j+1}. \tag{64}$$

Consequently, for every $\mathfrak{h} \ge 1$,

$$e_{\mathfrak{h}} = \sum_{j=0}^{\mathfrak{h}-1} \widehat{A}(g)^{\mathfrak{h}-1-j}\left[\delta_j(g) + W\xi_{j+1}\right]. \tag{65}$$

Proof. Subtract the predicted transition from the true transition:

$$e_{j+1} = A_W(g)z_j + C_W(g)\phi_j + W\xi_{j+1} - \widehat{A}(g)\widehat{z}_j. \tag{66}$$

Add and subtract $\widehat{A}(g)z_j$:

$$e_{j+1} = \widehat{A}(g)(z_j - \widehat{z}_j) + [A_W(g) - \widehat{A}(g)]z_j + C_W(g)\phi_j + W\xi_{j+1} \tag{67}$$

$$= \widehat{A}(g)e_j + \delta_j(g) + W\xi_{j+1}, \tag{68}$$

which proves Eq. 64. Since $e_0 = 0$, the first two steps are

$$e_1 = \delta_0 + W\xi_1, \tag{69}$$

$$e_2 = \widehat{A}(\delta_0 + W\xi_1) + \delta_1 + W\xi_2. \tag{70}$$

Repeating the substitution gives Eq. 65. The same conclusion also follows by induction on $\mathfrak{h}$. □

Since $\widehat{A}(g)^q = \widehat{S}_g(q)$, each one-step error is transformed by the remaining learned semigroup iterate. The terms are vectors, so they may reinforce or cancel one another. The identity does not imply that error, or the difference between two models, must grow monotonically with the horizon.

H.4.1 Physical readout and the central bound

Latent error and physical error are not the same quantity. The readout can hide some latent errors, and the representation can omit physical information even when its latent rollout is accurate.

Proposition H.11 (Exact physical readout decomposition)**.** *Let $s_\mathfrak{h} = C_{\text{phys}}\phi_\mathfrak{h}$ and let r be any frozen probe. Define the true-latent readout residual*

$$b_r(\phi_\mathfrak{h}) = s_\mathfrak{h} - r(z_\mathfrak{h}). \tag{71}$$

Then

$$s_\mathfrak{h} - r(\widehat{z}_\mathfrak{h}) = b_r(\phi_\mathfrak{h}) + r(z_\mathfrak{h}) - r(\widehat{z}_\mathfrak{h}). \tag{72}$$

The corresponding mean-squared error is

$$\mathbb{E}\|s_\mathfrak{h} - r(\widehat{z}_\mathfrak{h})\|_2^2 = \mathbb{E}\|b_r(\phi_\mathfrak{h})\|_2^2 + \mathbb{E}\|r(z_\mathfrak{h}) - r(\widehat{z}_\mathfrak{h})\|_2^2 + 2\mathbb{E}\left[b_r(\phi_\mathfrak{h})^\top\{r(z_\mathfrak{h}) - r(\widehat{z}_\mathfrak{h})\}\right]. \tag{73}$$

For the linear probe $r(z) = Rz$,

$$b_r(\phi_\mathfrak{h}) = (C_{\text{phys}} - RW)\phi_\mathfrak{h}, \qquad r(z_\mathfrak{h}) - r(\widehat{z}_\mathfrak{h}) = Re_\mathfrak{h}. \tag{74}$$

Proof. Add and subtract $r(z_\mathfrak{h})$:

$$s_\mathfrak{h} - r(\widehat{z}_\mathfrak{h}) = s_\mathfrak{h} - r(z_\mathfrak{h}) + r(z_\mathfrak{h}) - r(\widehat{z}_\mathfrak{h}). \tag{75}$$

Expanding the squared Euclidean norm and taking expectations gives Eq. 73. Substituting $r(z) = Rz$ gives Eq. 74. □

For the remainder of this subsection, specialize to $r(z) = Rz$ and write $\widehat{s}_\mathfrak{h} = R\widehat{z}_\mathfrak{h}$. For $\mathfrak{h} \geq 1$, define the physical rollout gain

$$\Gamma_\mathfrak{h}(g; R, \widehat{A}) = \sum_{q=0}^{\mathfrak{h}-1} \|R\widehat{A}(g)^q\|_2. \tag{76}$$

This quantity bounds how strongly the learned dynamics and readout can propagate a sequence of local latent errors.

Theorem H.12 (Pathwise physical error under an unseen gravity)**.** *Assume the shared law basis in Eq. 32. Suppose $\|z_j\|_2 \leq B_z$ and $\|\phi_j\|_2 \leq B_\phi$ along the test trajectory at $g_\star$. Then*

$$\|s_\mathfrak{h} - \widehat{s}_\mathfrak{h}\|_2 \leq \|(C_{\text{phys}} - RW)\phi_\mathfrak{h}\|_2 + \Gamma_\mathfrak{h}(g_\star; R, \widehat{A})\sqrt{\mathcal{L}_{\text{law}}(g_\star)}\,(B_z\epsilon_{\text{op}} + B_\phi\epsilon_{\text{cl}}) + \sum_{j=0}^{\mathfrak{h}-1}\left\|R\widehat{A}(g_\star)^{\mathfrak{h}-1-j}W\xi_{j+1}\right\|_2. \tag{77}$$

Proof. Proposition H.11 and Theorem H.10 give

$$s_{\mathfrak{h}} - \widehat{s}_{\mathfrak{h}} = (C_{\mathrm{phys}} - RW)\phi_{\mathfrak{h}} \tag{78}$$

$$+ \sum_{j=0}^{\mathfrak{h}-1} R\widehat{A}(g_\star)^{\mathfrak{h}-1-j}\delta_j(g_\star) \tag{79}$$

$$+ \sum_{j=0}^{\mathfrak{h}-1} R\widehat{A}(g_\star)^{\mathfrak{h}-1-j} W\xi_{j+1}. \tag{80}$$

Apply the triangle inequality. For the conditional-mean defect terms, Theorem H.4 gives the uniform bound

$$\|\delta_j(g_\star)\|_2 \le \sqrt{\mathcal{L}_{\mathrm{law}}(g_\star)}\,(B_z\epsilon_{\mathrm{op}} + B_\phi\epsilon_{\mathrm{cl}})\,. \tag{81}$$

Therefore

$$\sum_{j=0}^{\mathfrak{h}-1} \left\| R\widehat{A}(g_\star)^{\mathfrak{h}-1-j}\delta_j(g_\star)\right\|_2 \tag{82}$$

$$\le \sum_{j=0}^{\mathfrak{h}-1} \|R\widehat{A}(g_\star)^{\mathfrak{h}-1-j}\|_2 \|\delta_j(g_\star)\|_2 \tag{83}$$

$$\le \Gamma_{\mathfrak{h}}(g_\star; R, \widehat{A})\sqrt{\mathcal{L}_{\mathrm{law}}(g_\star)}\,(B_z\epsilon_{\mathrm{op}} + B_\phi\epsilon_{\mathrm{cl}})\,. \tag{84}$$

Keeping the transition-residual terms explicit proves Eq. 77. □

The bound separates the four quantities that affect physical rollout quality: the readout residual, the local unseen-law error, its recursive propagation, and the transition residuals. It is an upper bound, not an additive model of the observed error. The factor $\Gamma_{\mathfrak{h}}$ is a worst-case bound and does not describe cancellation between propagated residuals. Because the readout residual, trajectory bounds, rollout gain, and realized transition residuals remain in the statement, this is not a stand-alone generalization guarantee based only on the law-coverage factor.

H.4.2 Cross-step second moments

The orientation and cross-step dependence of the one-step errors also matter. Let

$$\varepsilon_j = \delta_j(g) + W\xi_{j+1}, \qquad S_{ij}(g) = \mathbb{E}[\varepsilon_i\varepsilon_j^\top \mid g]. \tag{85}$$

From Eq. 65, with $\widehat{A} = \widehat{A}(g)$,

$$\mathbb{E}[\|Re_{\mathfrak{h}}\|_2^2 \mid g] = \sum_{i=0}^{\mathfrak{h}-1}\sum_{j=0}^{\mathfrak{h}-1} \mathrm{tr}\Big(R\widehat{A}^{\mathfrak{h}-1-i} S_{ij}(g)(\widehat{A}^{\mathfrak{h}-1-j})^\top R^\top\Big)\,. \tag{86}$$

To derive this identity, write $Re_{\mathfrak{h}} = \sum_j R\widehat{A}^{\mathfrak{h}-1-j}\varepsilon_j$, expand the outer product, and take the trace. No zero-mean assumption is needed because $S_{ij}(g)$ is a second moment rather than a covariance. Its off-diagonal entries record cross-step dependence. A reset comparison changes these second moments as well as the propagation path, so it does not isolate a single spectral gain.

The same idea does not depend on linearity. If the learned nonlinear transition $\widehat{F}_g$ is L_g-Lipschitz on a set containing the true and predicted trajectories, and $d_j = \|\widehat{F}_g(x_j) - F_g(x_j)\|_2$, then

$$\|\widehat{x}_{\mathfrak{h}} - x_{\mathfrak{h}}\|_2 \le \sum_{j=0}^{\mathfrak{h}-1} L_g^{\mathfrak{h}-1-j} d_j. \tag{87}$$

Indeed, one step gives $\|\widehat{x}_{j+1} - x_{j+1}\|_2 \le L_g\|\widehat{x}_j - x_j\|_2 + d_j$, and iteration from a common initial state proves the claim. This nonlinear bound supplies intuition, but the exact linear recursion is the result used above.

H.5 WHY MULTI-STEP COMPOSITION CAN CHANGE THE LEARNED REPRESENTATION

The rollout horizon changes the geometry used to compare representations, even in a stable linear system. This section isolates the conditional-mean, action-free system and omits residual variation. The quadratic costs below therefore measure transition bias rather than irreducible noise. The finite-history version with actions appears in Section H.6.

Fix one smooth regime, suppress g, and let $\widehat{S}_A(\mathfrak{h}) = A^{\mathfrak{h}}$ for a candidate latent operator A. Define the finite-horizon semigroup-intertwining defect

$$D_{\mathfrak{h}}(W; A) = WS_g(\mathfrak{h}) - \widehat{S}_A(\mathfrak{h})W = WT^{\mathfrak{h}} - A^{\mathfrak{h}}W, \qquad D_1 = WT - AW. \tag{88}$$

When $A = \widehat{A}(g)$, $D_1\phi_t = \delta_t(g)$. When $A = A_W(g)$, $D_1 = C_W(g)$ is the representation closure residual. A general fitted A contains both closure and predictor error.

Theorem H.13 (Finite-horizon intertwining-defect geometry)**.** *For every* $\mathfrak{h} \geq 1$,

$$D_{\mathfrak{h}} = \sum_{j=0}^{\mathfrak{h}-1} A^{\mathfrak{h}-1-j} D_1 T^j. \tag{89}$$

Let

$$M_{\mathfrak{h}}(A, T) = \sum_{j=0}^{\mathfrak{h}-1} (T^j)^\top \otimes A^{\mathfrak{h}-1-j}. \tag{90}$$

For the feature second-moment matrix $\Sigma_\phi = \mathbb{E}[\phi\phi^\top] \succeq 0$ *and nonnegative weights* $w_1, \ldots, w_K$,

$$\sum_{k=1}^{K} w_k \operatorname{tr}(D_k \Sigma_\phi D_k^\top) = \operatorname{vec}(D_1)^\top G_K \operatorname{vec}(D_1), \tag{91}$$

where

$$G_K = \sum_{k=1}^{K} w_k M_k^\top (\Sigma_\phi \otimes I_{d_z}) M_k \succeq 0. \tag{92}$$

If $w_1 > 0$ *and* $\Sigma_\phi \succ 0$, *then* $G_K \succ 0$.

Proof. For $\mathfrak{h} = 1$, Eq. 89 is the definition of D_1. Suppose the identity holds at $\mathfrak{h}$. Then

$$\begin{aligned}
D_{\mathfrak{h}+1} &= WT^{\mathfrak{h}+1} - A^{\mathfrak{h}+1}W && (93)\\
&= (WT^{\mathfrak{h}} - A^{\mathfrak{h}}W)T + A^{\mathfrak{h}}(WT - AW) && (94)\\
&= D_{\mathfrak{h}}T + A^{\mathfrak{h}}D_1 && (95)\\
&= \sum_{j=0}^{\mathfrak{h}-1} A^{\mathfrak{h}-1-j} D_1 T^{j+1} + A^{\mathfrak{h}} D_1 && (96)\\
&= \sum_{j=0}^{\mathfrak{h}} A^{\mathfrak{h}-j} D_1 T^j. && (97)
\end{aligned}$$

Thus the telescoping identity holds by induction.

Using $\operatorname{vec}(LXR) = (R^\top \otimes L)\operatorname{vec}(X)$ in each term of Eq. 89 gives

$$\operatorname{vec}(D_{\mathfrak{h}}) = M_{\mathfrak{h}} \operatorname{vec}(D_1). \tag{98}$$

Also,

$$\operatorname{tr}(D_{\mathfrak{h}} \Sigma_\phi D_{\mathfrak{h}}^\top) = \operatorname{vec}(D_{\mathfrak{h}})^\top (\Sigma_\phi \otimes I_{d_z}) \operatorname{vec}(D_{\mathfrak{h}}). \tag{99}$$

Substitution and summation give Eq.s 91 and 92. Every summand in G_K is positive semidefinite. At $\mathfrak{h} = 1$, $M_1 = I$, so the first summand is $w_1(\Sigma_\phi \otimes I_{d_z})$, which is positive definite when $w_1 > 0$ and $\Sigma_\phi \succ 0$. □

The telescoping identity shows how a one-step failure of intertwining enters later compositions. If $D_1 = 0$, then $D_\mathfrak{h} = 0$ at every horizon. If $D_1 \neq 0$, the true and learned transition geometry determines whether the propagated terms reinforce or cancel. For fixed A, T, and Σ_ϕ, G_K weights these one-step defect directions by their later consequences. Two defects with the same one-step size can therefore have different multi-step costs. If fitting a new representation also changes A, one common G_K need not rank all representations.

Theorem H.14 (When training and evaluation weight defects differently)**.** *Let $G_{\mathrm{tr}} \succ 0$ and $G_{\mathrm{ev}} \succeq 0$ be the defect-weighting matrices for a training horizon profile and an evaluation profile on the same defect space. Set*

$$Q = G_{\mathrm{tr}}^{-1/2} G_{\mathrm{ev}} G_{\mathrm{tr}}^{-1/2}. \tag{100}$$

Then:

(a) If $G_{\mathrm{ev}} = cG_{\mathrm{tr}}$ for some $c \geq 0$, every defect has evaluation cost c times its training cost.

(b) If Q is not a scalar multiple of the identity, there are two defects with equal training cost and different evaluation cost.

(c) Let S be a nonzero subspace of the whitened coordinate y, let P_S be its orthogonal projector, and define the compressed operator $Q_S = P_S Q|_S$. If $Q_S \succ 0$, the largest ratio between evaluation costs of equal-training-cost defects whose whitened coordinates lie in S is $\lambda_{\max}(Q_S)/\lambda_{\min}(Q_S)$.

Proof. For a vectorized defect d, let $y = G_{\mathrm{tr}}^{1/2} d$. Its training cost is

$$d^\top G_{\mathrm{tr}} d = y^\top y, \tag{101}$$

and its evaluation cost is

$$d^\top G_{\mathrm{ev}} d = y^\top Q y. \tag{102}$$

If $G_{\mathrm{ev}} = cG_{\mathrm{tr}}$, then $Q = cI$, proving part (a). If Q is not scalar, choose unit eigenvectors y_1, y_2 with distinct eigenvalues. They have the same training cost but different evaluation costs, proving part (b). For $y \in S$, the evaluation cost is $y^\top Q_S y$. The Rayleigh quotient of Q_S ranges between its smallest and largest eigenvalues. Choosing the corresponding unit eigenvectors in S attains the ratio in part (c). If a subspace is specified in the original defect coordinate d, its image under $G_{\mathrm{tr}}^{1/2}$ is the corresponding subspace S in whitened coordinates. □

H.5.1 A STABLE RANKING REVERSAL

Explicit representations can make one-step and two-step profiling prefer opposite choices. The construction below is strictly stable, so the reversal does not rely on an unstable transition.

Theorem H.15 (One-step preference can reverse under rollout)**.** *Let $\phi_t \in \mathbb{R}^4$ have identity second moment and*

$$T = \mathrm{diag}\left(\frac{2}{5}, \frac{4}{5}, \frac{3}{20}, \frac{3}{5}\right). \tag{103}$$

Restrict the rank-one representation to

$$W_P = \frac{1}{\sqrt{2}}(1,1,0,0), \qquad W_F = \frac{1}{\sqrt{2}}(0,0,1,1), \tag{104}$$

and let the latent predictor be the scalar $A = a$. Then:

(a) the profiled one-step loss prefers W_P;

(b) using each representation's one-step-optimal predictor, the horizon-two loss prefers W_F;

(c) for the normalized objective

$$\frac{\|D_1\|_F^2 + \gamma \|D_2\|_F^2}{1+\gamma}, \tag{105}$$

full profiling prefers W_F whenever the sufficient condition $\gamma > 3600/9007 \approx 0.39969$, including $\gamma = 0.95$.

Proof. For a representation that averages two eigenmodes λ and μ,

$$\|D_{\mathfrak{h}}(W;a)\|_F^2 = \frac{1}{2}\left[(\lambda^{\mathfrak{h}} - a^{\mathfrak{h}})^2 + (\mu^{\mathfrak{h}} - a^{\mathfrak{h}})^2\right]. \tag{106}$$

At one step, differentiating with respect to a gives the minimizer $a = (\lambda+\mu)/2$ and minimum $(\lambda-\mu)^2/4$. Therefore

$$\min_a \|D_1(W_P;a)\|_F^2 = \frac{(\frac{4}{5} - \frac{2}{5})^2}{4} = \frac{1}{25}, \tag{107}$$

$$\min_a \|D_1(W_F;a)\|_F^2 = \frac{(\frac{3}{5} - \frac{3}{20})^2}{4} = \frac{81}{1600}. \tag{108}$$

Since $1/25 < 81/1600$, one-step profiling prefers W_P.

The one-step-optimal predictors are $a_P = 3/5$ and $a_F = 3/8$. At horizon two, Equation equation 106 gives

$$\|D_2(W_P;a_P)\|_F^2 = \frac{1}{2}\left[\left(\frac{4}{25} - \frac{9}{25}\right)^2 + \left(\frac{16}{25} - \frac{9}{25}\right)^2\right] = \frac{37}{625}, \tag{109}$$

$$\|D_2(W_F;a_F)\|_F^2 = \frac{1}{2}\left[\left(\frac{9}{400} - \frac{9}{64}\right)^2 + \left(\frac{9}{25} - \frac{9}{64}\right)^2\right] \tag{110}$$

$$= \frac{1}{2}\left[\left(-\frac{189}{1600}\right)^2 + \left(\frac{351}{1600}\right)^2\right] = \frac{79461}{2560000}. \tag{111}$$

Here $79461/2560000 < 37/625$, which proves the horizon-two reversal.

For the joint two-step objective, allow the horizon-one and horizon-two scalar predictions for W_P to vary independently. This relaxation can only lower its profiled cost, and it gives the lower bound

$$\inf_a \frac{\|D_1(W_P;a)\|_F^2 + \gamma\|D_2(W_P;a)\|_F^2}{1+\gamma} \geq \frac{\frac{1}{25} + \gamma\frac{36}{625}}{1+\gamma}. \tag{112}$$

For W_F, evaluate the admissible predictor $a = 2/5$. Its one-step and two-step costs are

$$\|D_1(W_F;2/5)\|_F^2 = \frac{41}{800}, \tag{113}$$

$$\|D_2(W_F;2/5)\|_F^2 = \frac{377}{12800}. \tag{114}$$

Hence its profiled cost is at most

$$\frac{\frac{41}{800} + \gamma\frac{377}{12800}}{1+\gamma}. \tag{115}$$

The upper bound in Eq. 115 is smaller than the lower bound in Eq. 112 when

$$\gamma\left(\frac{36}{625} - \frac{377}{12800}\right) > \frac{41}{800} - \frac{1}{25}, \tag{116}$$

$$\gamma\frac{9007}{320000} > \frac{3600}{320000}, \tag{117}$$

$$\gamma > \frac{3600}{9007}. \tag{118}$$

In that range, even the displayed upper bound for W_F is below a lower bound for W_P, so full profiling must prefer W_F. □

This construction proves that the objective can change representation preference as the rollout horizon changes. It does not prove that a longer rollout always helps, that a particular predictor architecture causes the change, or that this four-mode system is the mechanism learned by SG-JEPA.

H.6 Relating the theory to the history-based predictor

The preceding sections used a memoryless predictor to keep the mechanism visible. SG-JEPA instead reads a finite latent history together with actions and gravity. A companion state turns this interface into a one-step system on an augmented space, so the same algebra applies without treating the GRU as a literal matrix.

For context length H, define the feature-history state and its encoded form

$$X_t^{(H)} = \begin{bmatrix} \phi_{t-H+1} \\ \vdots \\ \phi_t \end{bmatrix} \in \mathbb{R}^{Hd_\phi}, \qquad \mathbb{W} = I_H \otimes W, \qquad Z_t^{(H)} = \mathbb{W} X_t^{(H)}. \tag{119}$$

Let $a_t \in \mathbb{R}^{d_a}$ contain every known exogenous action input needed for the transition from t to $t+1$. If the interface uses an action window, a_t denotes that fixed-dimensional stacked input. The true and predicted rollouts receive the same action sequence. Gravity remains fixed during the rollout and explicit through the law-indexed operators. Closed-loop action differences would introduce additional terms and are outside this prediction analysis.

For the nonlinear predictor, let $\widehat{\Phi}_{\theta,g}^{(k)}\left(Z_t^{(H)}; a_{t:t+k-1}\right)$ denote the predicted companion state obtained after the consecutive action block. Recursive application gives the control-sequence composition law

$$\begin{aligned} &\widehat{\Phi}_{\theta,g}^{(k+\ell)}\left(Z_t^{(H)}; a_{t:t+k+\ell-1}\right) \\ &\quad = \widehat{\Phi}_{\theta,g}^{(\ell)}\left(\widehat{\Phi}_{\theta,g}^{(k)}\left(Z_t^{(H)}; a_{t:t+k-1}\right); a_{t+k:t+k+\ell-1}\right). \end{aligned} \tag{120}$$

The predictor parameters are shared, but each numerical transition is indexed by its action input. In the action-free case, the companion update generates an ordinary discrete semigroup. With known actions, it obeys the controlled composition law above rather than an autonomous semigroup on an individual latent state.

Consider the linear companion systems

$$X_{t+1}^{(H)} = \mathbb{T}(g) X_t^{(H)} + \mathbb{B}(g) a_t + \mathbb{J}_\xi \xi_{t+1}, \tag{121}$$

$$\widehat{Z}_{t+1}^{(H)} = \mathbb{A}(g) \widehat{Z}_t^{(H)} + \mathbb{D}(g) a_t. \tag{122}$$

The first $H-1$ block rows of $\mathbb{T}(g)$ and $\mathbb{A}(g)$ shift the history. The final block row performs the true or learned update. The matrix $\mathbb{J}_\xi \in \mathbb{R}^{Hd_\phi \times d_\phi}$ inserts the new residual in the last feature block. The remaining dimensions are

$$\mathbb{T}(g) \in \mathbb{R}^{Hd_\phi \times Hd_\phi}, \qquad \mathbb{B}(g) \in \mathbb{R}^{Hd_\phi \times d_a}, \tag{123}$$

$$\mathbb{A}(g) \in \mathbb{R}^{Hd_z \times Hd_z}, \qquad \mathbb{D}(g) \in \mathbb{R}^{Hd_z \times d_a}. \tag{124}$$

The additive maps $\mathbb{B}(g)$ and $\mathbb{D}(g)$ are assumptions of this model. Nonlinear interactions among state, action, and gravity in the implemented predictor fall outside this additive model. For $1 \le L \le H$, let

$$P_{H,L} = \begin{bmatrix} 0_{Ld_z \times (H-L)d_z} & I_{Ld_z} \end{bmatrix} \in \mathbb{R}^{Ld_z \times Hd_z} \tag{125}$$

select the newest L latent blocks, and write $P_H = P_{H,1}$. At the start of rollout, both systems use the same true context:

$$\widehat{Z}_t^{(H)} = \mathbb{W} X_t^{(H)}. \tag{126}$$

For $\mathfrak{h} \ge 1$ and $0 \le j < \mathfrak{h}$, define

$$\mathfrak{D}_\mathfrak{h}(g) = P_H\left[\mathbb{W}\mathbb{T}(g)^\mathfrak{h} - \mathbb{A}(g)^\mathfrak{h}\mathbb{W}\right], \tag{127}$$

$$\mathfrak{F}_{\mathfrak{h},j}(g) = P_H\left[\mathbb{W}\mathbb{T}(g)^{\mathfrak{h}-1-j}\mathbb{B}(g) - \mathbb{A}(g)^{\mathfrak{h}-1-j}\mathbb{D}(g)\right], \tag{128}$$

$$\mathfrak{N}_{\mathfrak{h},j}(g) = P_H \mathbb{W}\mathbb{T}(g)^{\mathfrak{h}-1-j}\mathbb{J}_\xi. \tag{129}$$

Theorem H.16 (Exact finite-history rollout error)**.** *Under Eq.s 121 through 126, the newest latent error at horizon $\mathfrak{h}$ is*

$$z_{t+\mathfrak{h}} - \widehat{z}_{t+\mathfrak{h}} = \mathfrak{D}_{\mathfrak{h}}(g) X_t^{(H)} + \sum_{j=0}^{\mathfrak{h}-1} \mathfrak{F}_{\mathfrak{h},j}(g) a_{t+j} + \sum_{j=0}^{\mathfrak{h}-1} \mathfrak{N}_{\mathfrak{h},j}(g) \xi_{t+j+1}. \tag{130}$$

If $V_{t,\mathfrak{h}}$ stacks $X_t^{(H)}$, the $\mathfrak{h}$ future actions, and the $\mathfrak{h}$ transition residuals, and $\mathfrak{M}_{\mathfrak{h}}(g)$ is the corresponding block row in Equation equation 130, then

$$\mathbb{E}\big[\|z_{t+\mathfrak{h}} - \widehat{z}_{t+\mathfrak{h}}\|_2^2 \mid g\big] = \operatorname{tr}\big(\mathfrak{M}_{\mathfrak{h}}(g)\mathbb{E}[V_{t,\mathfrak{h}} V_{t,\mathfrak{h}}^\top \mid g]\mathfrak{M}_{\mathfrak{h}}(g)^\top\big)\,. \tag{131}$$

No centering or independence assumption is required.

Proof. Repeated substitution in the true companion system gives

$$X_{t+\mathfrak{h}}^{(H)} = \mathbb{T}(g)^{\mathfrak{h}} X_t^{(H)} \tag{132}$$

$$+ \sum_{j=0}^{\mathfrak{h}-1} \mathbb{T}(g)^{\mathfrak{h}-1-j} \left[\mathbb{B}(g) a_{t+j} + \mathbb{J}_\xi \xi_{t+j+1}\right]. \tag{133}$$

The exact initial context and recursive predictor give

$$\widehat{Z}_{t+\mathfrak{h}}^{(H)} = \mathbb{A}(g)^{\mathfrak{h}} \mathbb{W} X_t^{(H)} \tag{134}$$

$$+ \sum_{j=0}^{\mathfrak{h}-1} \mathbb{A}(g)^{\mathfrak{h}-1-j} \mathbb{D}(g) a_{t+j}. \tag{135}$$

The true newest latent is $P_H \mathbb{W} X_{t+\mathfrak{h}}^{(H)}$, and the predicted newest latent is $P_H \widehat{Z}_{t+\mathfrak{h}}^{(H)}$. Apply these selectors to Eq.s 133 and 135, subtract, and group the context, action, and transition-residual terms. Their coefficients are exactly $\mathfrak{D}_{\mathfrak{h}}$, $\mathfrak{F}_{\mathfrak{h},j}$, and $\mathfrak{N}_{\mathfrak{h},j}$.

For the second statement, write the error as $\mathfrak{M}_{\mathfrak{h}}(g) V_{t,\mathfrak{h}}$. Then

$$\mathbb{E}\|\mathfrak{M}_{\mathfrak{h}} V_{t,\mathfrak{h}}\|_2^2 = \mathbb{E}\operatorname{tr}\big(\mathfrak{M}_{\mathfrak{h}} V_{t,\mathfrak{h}} V_{t,\mathfrak{h}}^\top \mathfrak{M}_{\mathfrak{h}}^\top\big) \tag{136}$$

$$= \operatorname{tr}\big(\mathfrak{M}_{\mathfrak{h}} \mathbb{E}[V_{t,\mathfrak{h}} V_{t,\mathfrak{h}}^\top] \mathfrak{M}_{\mathfrak{h}}^\top\big)\,. \tag{137}$$

Conditioning throughout on g proves Eq. 131. □

Theorem H.16 is stated for the newest latent to match the rollout loss. Replacing P_H by $P_{H,L}$ in Eq.s 127–129 gives the same exact identity for the stacked window $z_{t+\mathfrak{h}-L+1:t+\mathfrak{h}} - \widehat{z}_{t+\mathfrak{h}-L+1:t+\mathfrak{h}}$. A linear probe on that window is then represented by a matrix multiplying $P_{H,L}$.

Corollary H.17 (Memoryless results on the companion state)**.** *Suppose the augmented true and learned operators share the law basis used in Section H.3. Then the local-defect, law-transfer, and recursive-error results apply on the companion state after the substitutions*

$$W \mapsto \mathbb{W}, \qquad T(g) \mapsto \mathbb{T}(g), \qquad \widehat{A}(g) \mapsto \mathbb{A}(g), \tag{138}$$

followed by the selector P_H. For the horizon-geometry result, define

$$\mathbb{D}_{\mathfrak{h}} = \mathbb{W}\mathbb{T}^{\mathfrak{h}} - \mathbb{A}^{\mathfrak{h}}\mathbb{W},$$
$$\mathbb{M}_{\mathfrak{h}} = \sum_{j=0}^{\mathfrak{h}-1} (\mathbb{T}^j)^\top \otimes \mathbb{A}^{\mathfrak{h}-1-j}, \qquad \Sigma_X = \mathbb{E}[X_t^{(H)} X_t^{(H)\top}]. \tag{139}$$

The quadratic form for newest-block loss uses

$$G_K^{(P)} = \sum_{k=1}^{K} w_k \mathbb{M}_k^\top \left(\Sigma_X \otimes P_H^\top P_H\right) \mathbb{M}_k \succeq 0. \tag{140}$$

For a linear L-block history probe, replace P_H by $P_{H,L}$ in this quadratic form and compose the result with the probe matrix. This matrix need not be positive definite on every augmented defect

because P_H ignores the older output blocks. If $w_1 > 0$ and $\Sigma_X \succ 0$, it is positive definite after restriction to the admissible companion-defect space

$$\mathcal{S}_{\mathrm{adm}} = \left\{ \mathrm{vec}\left(\begin{bmatrix} 0_{(H-1)d_z \times H d_\phi} \\ D \end{bmatrix} \right) : D \in \mathbb{R}^{d_z \times H d_\phi} \right\}. \tag{141}$$

The zero rows are the matched history shifts. If the true and learned action maps differ, the action residuals $\mathfrak{F}_{\mathfrak{h},j}(g)a_{t+j}$ remain as the explicit extra terms in Eq. 130. The shared basis must include an intercept to represent the gravity-independent shift rows of the companion matrices.

Proof. Eq.s 121 and 122 have the same algebraic form as the memoryless systems on their augmented spaces. Applying the earlier results therefore produces augmented latent errors. Multiplication by P_H selects the output that the model evaluates. The action terms vanish only when their true and learned augmented maps agree; otherwise the exact difference is $\mathfrak{F}_{\mathfrak{h},j}(g)a_{t+j}$. For horizon geometry, $\mathrm{vec}(\mathbb{D}_{\mathfrak{h}}) = \mathbb{M}_{\mathfrak{h}} \mathrm{vec}(\mathbb{D}_1)$. Selecting the newest block changes the output metric from the identity to $P_H^\top P_H$, which gives Eq. 140. Its $\mathfrak{h} = 1$ term is strictly positive for every nonzero defect in $\mathcal{S}_{\mathrm{adm}}$ when $w_1 > 0$ and $\Sigma_X \succ 0$: P_H retains its only potentially nonzero block row, and Σ_X is positive definite. □

The companion construction is an exact bridge for the history-state model. It does not assert that the nonlinear GRU is globally linear. It states what the model must contain to represent history, known actions, gravity, and recursive feedback without dropping terms.

H.7 Regime changes and contact

A time-homogeneous hybrid map can still generate a discrete semigroup on the full state. Contact changes the active local branch, not the underlying update rule, so the branch matrices may vary along a trajectory. The smooth law-transfer analysis applies within one regime, while the telescoping identity below applies along a fixed, shared branch sequence. It does not cover errors that change the event time or send the prediction onto a different branch. The later hinge result concerns one continuous piecewise-affine boundary relative to an affine feature dictionary.

Let the true transitions along a path be $T_0, \ldots, T_{\mathfrak{h}-1}$ and the latent transitions be $A_0, \ldots, A_{\mathfrak{h}-1}$. Define the branch-local transition defects

$$\Delta_j^{\mathrm{br}} = WT_j - A_j W. \tag{142}$$

Lemma H.18 (Telescoping identity across changing regimes)**.** *With an empty product equal to the identity,*

$$\begin{aligned} &WT_{\mathfrak{h}-1} \cdots T_0 - A_{\mathfrak{h}-1} \cdots A_0 W \\ &\quad = \sum_{j=0}^{\mathfrak{h}-1} A_{\mathfrak{h}-1} \cdots A_{j+1} \Delta_j^{\mathrm{br}} T_{j-1} \cdots T_0. \end{aligned} \tag{143}$$

Proof. For each $j = 0, \ldots, \mathfrak{h}$, define

$$Q_j = A_{\mathfrak{h}-1} \cdots A_j W T_{j-1} \cdots T_0, \tag{144}$$

where $Q_{\mathfrak{h}} = WT_{\mathfrak{h}-1} \cdots T_0$ and $Q_0 = A_{\mathfrak{h}-1} \cdots A_0 W$. Consecutive terms satisfy

$$\begin{aligned} Q_{j+1} - Q_j &= A_{\mathfrak{h}-1} \cdots A_{j+1} (WT_j - A_j W) T_{j-1} \cdots T_0 && (145) \\ &= A_{\mathfrak{h}-1} \cdots A_{j+1} \Delta_j^{\mathrm{br}} T_{j-1} \cdots T_0. && (146) \end{aligned}$$

Summing $Q_{j+1} - Q_j$ from $j = 0$ to $\mathfrak{h} - 1$ cancels all intermediate Q_j and yields Eq. 143. □

The identity has the same interpretation as the fixed-regime telescope, but the operators before and after each local defect now depend on the active branch.

Theorem H.19 (A continuous piecewise-affine event requires a hinge)**.** *Let an open connected set $U \subset \mathbb{R}^d$ intersect both sides of the hyperplane $a^\top x = b$, with $a \neq 0$. Suppose a continuous map $F : U \to \mathbb{R}^m$ is affine on each side:*

$$F(x) = \begin{cases} A_- x + c_-, & a^\top x \leq b, \\ A_+ x + c_+, & a^\top x \geq b. \end{cases} \tag{147}$$

Then there is a vector $u \in \mathbb{R}^m$ such that

$$F(x) = A_- x + c_- + u(a^\top x - b)_+, \tag{148}$$

where $(r)_+ = \max\{r, 0\}$. If $u \neq 0$, the hinge $(a^\top x - b)_+$ is not in the affine feature dictionary on U, so affine features alone cannot represent F. Appending this hinge is sufficient. In addition, any feature dictionary that contains the affine functions and admits an exact linear readout of F must contain the hinge in its linear span.

Proof. Choose a point $x_0 \in U$ on the boundary $a^\top x_0 = b$. Such a point exists because U is connected and intersects both open half-spaces. Let $v \in \ker(a^\top)$. For sufficiently small τ, both $x_0 + \tau v$ and $x_0 - \tau v$ remain on the boundary and in U. Continuity of the two affine formulas on the boundary gives

$$(A_+ - A_-)x + (c_+ - c_-) = 0 \tag{149}$$

for every boundary point x in a neighborhood of x_0. Subtract this identity at $x_0 + \tau v$ and x_0 to obtain

$$(A_+ - A_-)v = 0 \qquad \text{for every } v \in \ker(a^\top). \tag{150}$$

Each row of $A_+ - A_-$ therefore annihilates the $(d-1)$-dimensional space $\ker(a^\top)$. Its row space lies in the one-dimensional span of $a^\top$, so some $u \in \mathbb{R}^m$ satisfies

$$A_+ - A_- = ua^\top. \tag{151}$$

Evaluate the boundary identity at x_0. Since $a^\top x_0 = b$,

$$0 = ua^\top x_0 + c_+ - c_- = ub + c_+ - c_-, \tag{152}$$

and hence $c_+ - c_- = -ub$. On the upper half-space,

$$A_+ x + c_+ = A_- x + c_- + u(a^\top x - b), \tag{153}$$

while the added term is zero on the lower half-space. This proves Eq. 148.

If $u \neq 0$ and the hinge were affine on U, it would vanish as an affine function on the nonempty open set $U \cap \{a^\top x < b\}$. Its coefficients would then be zero, forcing it to vanish throughout U. This contradicts its nonzero values on $U \cap \{a^\top x > b\}$. Thus an affine dictionary cannot represent the hinge. Adding the displayed hinge is sufficient by Eq. 148. Finally, because $u \neq 0$, choose $q \in \mathbb{R}^m$ with $q^\top u = 1$. Eq. 148 then gives

$$(a^\top x - b)_+ = q^\top F(x) - q^\top (A_- x + c_-). \tag{154}$$

If a dictionary contains affine functions and linearly represents F, the right-hand side lies in its linear span. The dictionary must therefore span the hinge, which completes the proof. □

Corollary H.20 (Ballistic contact creates a gravity hinge)**.** *Suppose the free-flight height at a fixed time is $y_{\text{free}}(g) = a - bg$ with $b > 0$, and a floor clips the height to*

$$y(g) = \max\{a - bg, 0\}. \tag{155}$$

Let $g_c = a/b$. On any interval that crosses g_c,

$$y(g) = b(g_c - g)_+ \tag{156}$$

is not affine in g.

Proof. For $g < g_c$, the height is $a - bg = b(g_c - g)$ and has derivative $-b$. For $g > g_c$, the height is zero and has derivative zero. The derivative changes at g_c, so no single affine function agrees on an interval crossing that point. The displayed hinge agrees with both branches. □

The result covers one continuous piecewise-affine boundary. It does not cover discontinuous, frictional, restitution-dependent, or multi-contact dynamics, nor does it imply a literal neural hinge, a larger latent for every collision, or better contact modeling by SG-JEPA. It affects latent closure only if the retained quantity responds to the slope change u.

H.8 SUPPLEMENTARY CALCULATION: SIGREG AND ITS LIMITS

SIGReg enters Equation equation 13, but it is separate from the composition and law-coverage arguments. Near Gaussian identity covariance, its population statistic penalizes scale mismatch and unequal directional variance, while a finite batch introduces bias. This local calculation neither proves noncollapse nor attributes the dynamics difference to SIGReg.

For a batch of $B \geq 1$ latent samples $z_1, \ldots, z_B \in \mathbb{R}^{d_z}$, a random direction $a \sim \mathrm{Unif}(\mathbb{S}^{d_z-1})$, knots t_ℓ, and weights $\omega_\ell \geq 0$, consider

$$\mathcal{R}_{\mathrm{SIG}} = B \sum_\ell \omega_\ell \left| \frac{1}{B} \sum_{i=1}^{B} e^{\mathrm{i} t_\ell a^\top z_i} - e^{-t_\ell^2/2} \right|^2 . \tag{157}$$

Here $\mathrm{i}^2 = -1$. The batch average is the empirical characteristic function, a Fourier summary of the distribution after projection onto a. For a zero-mean Gaussian with covariance C, define the direction-averaged population functional

$$\mathcal{R}^{\mathrm{pop}}_{\mathrm{SIG}}(C) = \mathbb{E}_a \left[B \sum_\ell \omega_\ell \left| \mathbb{E}_{z\sim\mathcal{N}(0,C)} e^{\mathrm{i} t_\ell a^\top z} - e^{-t_\ell^2/2} \right|^2 \right]. \tag{158}$$

Proposition H.21 (Local covariance penalty induced by SIGReg)**.** *Assume some nonzero knot has positive weight. If $C = I + \Delta_C \succ 0$, $\Delta_C = \Delta_C^\top$, and $\|\Delta_C\|_F$ is small, then the functional in Eq. 158 satisfies*

$$\mathcal{R}^{\mathrm{pop}}_{\mathrm{SIG}}(I + \Delta_C) = \kappa_{B,d_z} \left[2\|\Delta_C\|_F^2 + (\operatorname{tr} \Delta_C)^2 \right] + O(\|\Delta_C\|_F^3), \tag{159}$$

$$\kappa_{B,d_z} = \frac{B}{d_z(d_z+2)} \sum_\ell \omega_\ell \frac{t_\ell^4}{4} e^{-t_\ell^2}. \tag{160}$$

For a finite batch of independent samples and a fixed direction with $v = a^\top C a$, the exact expectation at one knot is

$$B \left(e^{-t^2 v/2} - e^{-t^2/2} \right)^2 + 1 - e^{-t^2 v}. \tag{161}$$

Consequently, after averaging over both the independent batch and the random direction, for a constant C_0 that does not depend on Δ_C,

$$\begin{aligned} \mathbb{E}\mathcal{R}^{\mathrm{emp}}_{\mathrm{SIG}}(I + \Delta_C) = C_0 &+ \beta_{1,d_z} \operatorname{tr} \Delta_C \\ &+ \kappa^{\mathrm{emp}}_{B,d_z} \left[2\|\Delta_C\|_F^2 + (\operatorname{tr} \Delta_C)^2 \right] + O(\|\Delta_C\|_F^3), \end{aligned} \tag{162}$$

$$\beta_{1,d_z} = \frac{1}{d_z} \sum_\ell \omega_\ell t_\ell^2 e^{-t_\ell^2},$$

$$\kappa^{\mathrm{emp}}_{B,d_z} = \frac{B-2}{d_z(d_z+2)} \sum_\ell \omega_\ell \frac{t_\ell^4}{4} e^{-t_\ell^2}.$$

Proof. For fixed a, the scalar $a^\top z$ is Gaussian with variance $v = a^\top C a$. Its characteristic function at t is $e^{-t^2 v/2}$. Write $v = 1 + \eta$, where $\eta = a^\top \Delta_C a$. A Taylor expansion around $\eta = 0$ gives

$$e^{-t^2(1+\eta)/2} - e^{-t^2/2} = e^{-t^2/2} \left(e^{-t^2\eta/2} - 1 \right) \tag{163}$$

$$= -\frac{t^2}{2} e^{-t^2/2} \eta + O(\eta^2), \tag{164}$$

$$\left(e^{-t^2(1+\eta)/2} - e^{-t^2/2} \right)^2 = \frac{t^4}{4} e^{-t^2} \eta^2 + O(|\eta|^3). \tag{165}$$

For a uniform direction on the unit sphere,

$$\mathbb{E}_a[\eta] = \frac{\operatorname{tr} \Delta_C}{d_z}, \tag{166}$$

$$\mathbb{E}_a[\eta^2] = \frac{2\|\Delta_C\|_F^2 + (\operatorname{tr} \Delta_C)^2}{d_z(d_z+2)}. \tag{167}$$

Multiplying by $B\omega_\ell$, summing over knots, and using the second identity proves Eq. 159.

For the finite batch, let $Y_i = e^{\mathrm{i}ta^\top z_i}$. Since $|Y_i| = 1$,

$$\mathbb{E}\left|\frac{1}{B}\sum_{i=1}^{B} Y_i - e^{-t^2/2}\right|^2 = \left|e^{-t^2 v/2} - e^{-t^2/2}\right|^2 \tag{168}$$

$$+ \frac{1}{B}\left(1 - e^{-t^2 v}\right). \tag{169}$$

Multiplication by B gives Eq. 161. Finally,

$$1 - e^{-t^2(1+\eta)} = 1 - e^{-t^2} + t^2 e^{-t^2}\eta - \frac{t^4}{2}e^{-t^2}\eta^2 + O(|\eta|^3). \tag{170}$$

Combining this with the population mismatch, then applying both identities in Eq. 167, gives the stated values of β_{1,d_z} and $\kappa^{\mathrm{emp}}_{B,d_z}$. □

Because at least one nonzero knot has positive weight, $\kappa_{B,d_z} > 0$ and the population functional has a strict local quadratic minimum at identity covariance. The expected finite-batch statistic also has a linear trace term, so identity covariance need not minimize it. When $B > 2$, $\kappa^{\mathrm{emp}}_{B,d_z} > 0$ as well, but the linear term remains. The expansion supports SIGReg's local scale and isotropy role. It does not analyze collapse at $C = 0$, connect effective rank to prediction, or explain the representation difference.