Unlocking Temporal Generalization in Hamiltonian Video Dynamics Models

Eli Laird[1][0000−0002−0668−8745] and Corey Clark[1][0000−0003−0129−9270]

Department of Computer Science, Southern Methodist University,
Dallas, TX, USA
{ejlaird,coreyc}@smu.edu

Abstract. World models are typically trained to predict discrete-time physical dynamics with a fixed step size baked into the model weights, preventing prediction at variable temporal resolutions. This matters for hierarchical planning, sim-to-real transfer, and scientific or game-engine applications that must query the same dynamics at multiple timescales. Hamiltonian Generative Networks (HGN) offer a principled path forward, grounding predictions in a continuous-time energy function that is, in principle, independent of the observation frame rate. In practice, however, their temporal generalization breaks down in non-conservative settings. We show that in externally forced, dissipative environments, HGN rollouts at step sizes beyond the training regime fail due to distinct failure modes, including latent magnitude growth driven by an unconstrained action-force map, and global truncation error accumulation from an under-resolved integrator. We identify a targeted fix for each mechanism and demonstrate stable dynamics prediction at temporal resolutions well outside the training distribution. In a detailed analysis, we recommend several strategies for enabling temporal generalization in continuous-time video generation.

Keywords: Physics-aware Video Generation · Physics-informed Neural Networks · Temporal Generalization · Hamiltonian Dynamics

1 Introduction

Most world model architectures predict physical dynamics on a discrete time grid with a fixed step size at training time. The dominant paradigm in world modeling is a discrete-time transition $z_{t+1} = f(z_t, a_t)$, the structure underlying the early World Model architectures [32,12,11], the Dreamer series [14,15,16], the JEPA-based models [19,4,35,20], and others [1,29,10,23]. These models learn capable predictors at their training rate but have no mechanism to generalize across temporal resolutions, which matters for hierarchical planning (coarse and fine time scales) [26,13,34], sim-to-real transfer (simulation and control rates rarely match) [22,27], and scientific simulation engines that must query the same dynamics at multiple timescales [5].

Neural ODEs [6,24] address the fixed-step constraint by parameterizing dynamics as a vector field $\frac{dz}{dt} = f(z)$ and integrating at any desired step size. But

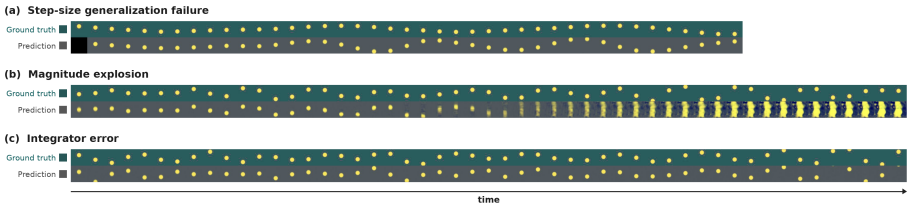

Fig. 1: Three views of how dynamics rollouts can fail at evaluation step sizes beyond the training step. In each panel the top strip (teal) is the ground-truth frame sequence and the bottom strip (gray) is the model's open-loop prediction. **(a)** *Step-size generalization failure*: past the training step the predicted rollout defaults to the memorized dynamics at the training step size. **(b)** *Magnitude explosion*: an unconstrained integrator injects energy causing the rendered mass smears into high-amplitude artifacts. **(c)** *Integrator error*: with the energy-explosion constrained, the rollout stays bounded but accumulates global truncation error from the under-resolved integrator, drifting out of phase with the ground truth.

the learned field is an unconstrained network, which without physical inductive biases may not respect conservation laws or symplectic structure, destabilizing long-horizon rollouts.

Hamiltonian neural networks provide exactly this conservative and symplectic structure [9]. The Hamiltonian is a scalar energy function $H(q, p)$ with gradients that define the equations of motion and guarantee energy conservation, time reversibility, and symplectic flow. Hamiltonian Generative Networks (HGN) [28] demonstrated plausible visual rollouts at double and half the training rate from a single model, as well as time-reversed dynamics. Yet in non-conservative settings this temporal generalization degrades, and its limits have not been systematically characterized.

We study temporal generalization in Hamiltonian Generative Networks through the lens of interpolation and extrapolation, predicting at step sizes, respectively, smaller and larger than training. Interpolation generalizes naturally, since the symplectic integrator refines any target step without retraining; extrapolation, where the step exceeds anything seen during training, is more fragile and is the focus of this work (Section 3).

We extend HGN with port-Hamiltonian structure to handle externally forced, dissipative environments and identify multiple failure modes where rollouts fail beyond the training regime, including exponential latent magnitude growth driven by an unconstrained action-force map, and global truncation error accumulation from an under-resolved integrator (Fig. 1). We provide a targeted fix for each and suggest a strategy for learning visual continuous-time dynamics at multiple temporal resolutions.

2 Related Work

2.1 World Models

Most world models predict environment dynamics as a discrete-time latent transition $z_{t+1} = f(z_t, a_t)$ learned at a single training frame rate. Early latent dynamics models established this template directly from pixels. Embed to Control [32] learned a locally linear latent transition for control from raw images, and the World Models of Ha and Schmidhuber [12,11] paired a variational autoencoder with a recurrent mixture-density network that rolls the latent state forward one step at a time. The Dreamer line scaled this recipe. Dreamer [14] learns behaviors by latent imagination inside a recurrent state-space model, DreamerV2 [15] replaced the continuous latent with discrete categorical codes, and DreamerV3 [16] stabilized the approach across diverse domains. In every case the recurrent update advances the state by exactly one fixed environment step, with no notion of the underlying continuous time.

A second family predicts in a learned representation space rather than in pixels. The joint-embedding predictive architecture (JEPA) proposed by LeCun [19], with image-level instantiations such as I-JEPA [2] and the collapse-free LeJEPA [3], motivates predicting future embeddings instead of reconstructing observations. Built on top of these JEPA encoders, DINO-WM [35] learns next-step dynamics over frozen DINO features to enable zero-shot planning, Navigation World Models [4] predict future egocentric observations conditioned on navigation actions, and LeWorldModel [20] trains a JEPA next-embedding predictor end-to-end from pixels. These predictors are likewise autoregressive over a fixed step and cannot be queried at an arbitrary temporal resolution.

Generative video models form a third family: DIAMOND [1] learns a diffusion world model that preserves fine visual detail for agent training, GameNGen [29] turns a diffusion next-frame predictor into a real-time neural game engine, and Long-Context State-Space Video World Models [23] adopt state-space backbones to extend the memory of autoregressive frame prediction. A few models reason across multiple time scales, but only over integer multiples of the base step. THICK [10] learns a hierarchy in which a high level predicts sparse context-change events above a discrete low-level latent dynamics. Across all of these families the predictor is bound to the temporal resolution of its training data, advancing by a fixed step rather than integrating a continuous-time vector field.

2.2 Neural ODEs and Continuous-time Dynamics in State Space

A parallel line of work, focused on Neural ODEs, has embedded physics into learned dynamics through architectural inductive biases. Hamiltonian Neural Networks [9] learned a scalar Hamiltonian $H(q, p)$ from state observations, demonstrating energy conservation on simple systems. Lagrangian Neural Networks [7] took the dual approach in position-velocity space. Neural ODEs [6] parameterized continuous-time dynamics via ODE solvers, making the timestep a free parameter, and Latent ODEs [24] extended this to irregularly-sampled observations in

latent space. Port-Hamiltonian neural networks [8] built on the port-Hamiltonian formalism [25] to handle dissipative systems with external forcing. All of these methods operate in known low-dimensional state spaces, while our model embeds the port-Hamiltonian structure within a pixel-to-pixel generative pipeline, introducing the additional challenge of jointly learning the visual state representation and the dynamics.

2.3 Hamiltonian Generative Networks

The approach most directly comparable to ours is Hamiltonian Generative Networks (HGN) [28], which first learned the Hamiltonian dynamics end-to-end from pixel observations. HGN infers a latent phase-space state, integrates a learned separable Hamiltonian with a symplectic leapfrog integrator, and decodes images from latent "position" coordinates. The model is trained with a temporally extended ELBO objective [18] and demonstrated forward, backward, $2\times$, and $\frac{1}{2}\times$ speed rollouts from a single model by changing the magnitude of the integration step. Our work extends HGN with a port-Hamiltonian external force (action) injection and characterizes when and why temporal generalization breaks down at larger step-size ratios.

3 Interpolation and Extrapolation

Temporal generalization has two directions that test fundamentally different things about what a model has learned. *Interpolation* refers to evaluation at step sizes smaller than the training step. In this setting, the model samples its learned trajectory more densely, and the symplectic integrator can decompose any target step into finer substeps without retraining. Strong interpolation performance, however, does not distinguish a model that learned continuous-time physics from one that memorized a smooth curve. *Extrapolation* refers to evaluation at step sizes larger than training, where the integration step exceeds anything the model encountered during learning. A model that learned a genuine continuous-time vector field should fail for numerical reasons including the solver step being too large and producing truncation error that finer integration can remove. A model that memorized a fixed-step transition map should fail for representational reasons. These fixed-step models simply reproduce the dynamics at the training step size, and refining the integration cannot help. This asymmetry makes extrapolation a useful test for the continuous-time structure and is the focus of the analysis that follows.

4 Methods

Our model follows the HGN design [28] in which an encoder infers the dynamical state from pixels, a separable Hamiltonian network integrates the dynamics in an abstract phase space, and a decoder reconstructs images from the integrated

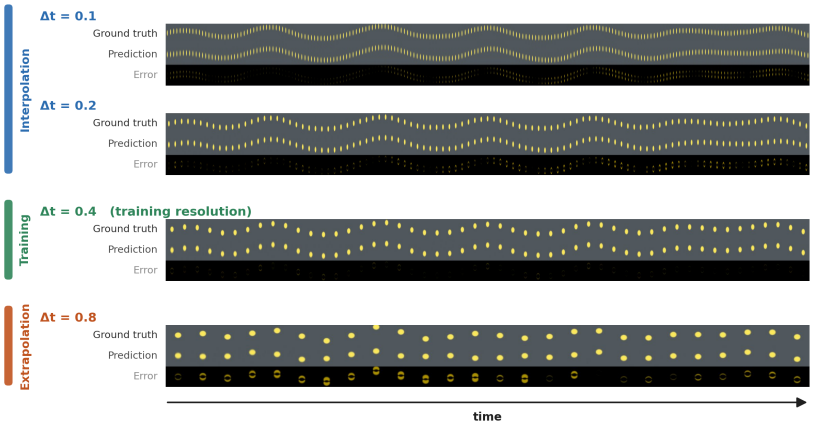

Fig. 2: Temporal generalization of the baseline port-HGN. Each block renders the same underlying oscillator trajectory sampled at a different evaluation step size Δt_{eval}, shown as ground truth (top), prediction (middle), and prediction error (bottom). Below the training step the model interpolates accurately ($\Delta t_{\text{eval}} \in \{0.1, 0.2\}$) and it reproduces the training resolution ($\Delta t_{\text{eval}} = \Delta t_{\text{train}} = 0.4$), with error rows that remain near-black. In extrapolation ($\Delta t_{\text{eval}} = 0.8$) the error row develops pronounced residuals as the prediction drifts out of phase.

state. We extend this with port-Hamiltonian structure, adding energy dissipation and action forcing.

Encoder and Decoder. A convolutional network with four residual blocks and stride-2 downsampling (64 channels) followed by a 2-layer MLP maps each 64×64×3 frame to a d=64 dimensional latent vector. The latent is partitioned equally into abstract position q and momentum p. A convolutional upsampling network reconstructs images from position coordinates only: $\hat{x}_t = D_\rho(q_t)$, following Toth et al. [28].

Port-Hamiltonian predictor. Given state (q, p) and action a, the predictor integrates the port-Hamiltonian equations

$$\dot{q} = \frac{\partial H}{\partial p}, \qquad \dot{p} = -\frac{\partial H}{\partial q} - \gamma \frac{\partial H}{\partial p} + G(a), \tag{1}$$

where $H(q, p) = T(p) + V(q)$ is a learned separable Hamiltonian with T and V each parameterized as scalar-output MLPs with Softplus activations, $\gamma \geq 0$ is a learned scalar damping coefficient, and $G(a)$ is a learned action-force map.

Leapfrog Integrator. The physics integrator is a modified leapfrog [30] that folds the damping and forcing symmetrically into both momentum half-steps:

$$\begin{aligned} p_{\text{half}} &= p_t + \tfrac{\Delta t}{2}\left(-\left.\frac{\partial H}{\partial q}\right|_t - \gamma \left.\frac{\partial H}{\partial p}\right|_t + G(a)\right) \\ q_{t+1} &= q_t + \Delta t \left.\frac{\partial H}{\partial p}\right|_{\text{half}} \\ p_{t+1} &= p_{\text{half}} + \tfrac{\Delta t}{2}\left(-\left.\frac{\partial H}{\partial q}\right|_{t+1} - \gamma \left.\frac{\partial H}{\partial p}\right|_{\text{half}} + G(a)\right) \end{aligned} \tag{2}$$

The action force $G(a)$ is held constant across both half-steps, so the per-step impulse is exactly $\Delta t \cdot G$ regardless of substepping level. The leapfrog integrator is conditionally stable, meaning, for an oscillator of frequency ω, the update is stable only for $\Delta t < 2/\omega$.

Objective. All models are trained with the variational objective of Toth et al. [28]. The encoder maps the context frames to a posterior $q_\phi(z \mid x_{0:T})$ over the initial latent state. A reparameterized sample is then projected to the initial phase-space state (q_0, p_0), which is then rolled forward for the full horizon by the integrator without re-encoding or teacher forcing. Every predicted position q_t in the trajectory is decoded independently, and the loss is

$$\mathcal{L}(\phi, \theta) = \frac{1}{T+1}\sum_{t=0}^{T} \mathbb{E}_{q_\phi(z|x_{0:T})}[\log p_\theta(x_t \mid q_t)] - \beta_{\text{KL}} \cdot KL(q_\phi(z \mid x_{0:T}) \,\|\, p(z)), \tag{3}$$

where ϕ denotes the encoder parameters, θ the dynamics and decoder parameters, the first term is the per-frame reconstruction likelihood averaged over the rollout horizon, and the second is the KL divergence of the posterior against an isotropic unit Gaussian prior with $\beta_{\text{KL}} = 1$. Because the entire trajectory is generated from a single initial encode, the reconstruction term is an open-loop rollout loss, i.e., the error at frame t reflects t accumulated integration steps. The decoder operates on position coordinates only which means momentum drives the dynamics but is never directly decoded. All models are optimized with Adam (learning rate 5×10^{-4} to 5×10^{-5}, cosine schedule, and trained to convergence).

Inference-time substepping. To discriminate integrator error from representational failure, we introduce a inference-time substepping protocol. Given an observation step Δt, we split each step into N integration substeps of size $\Delta t/N$, applying the action force at each substep. The total action impulse is preserved ($N \cdot (\Delta t/N) \cdot G = \Delta t \cdot G$), so substepping affects only the integration resolution and not the action magnitude. The encoder, learned Hamiltonian, and inferred initial state are identical across all substepping conditions, therefore, any performance change is attributable to the reduction in truncation error.

5 Experimental Setup

Environment. We evaluate on a forced damped mass-spring oscillator rendered as 64×64 RGB pixel observations, with mass $m = 0.5$, spring constant $k = 2.0$,

and damping coefficient $c = 0.1$, giving natural frequency $\omega = 2\,\mathrm{rad/s}$ and period $T_{\mathrm{period}} \approx 3.14$ time units. The system is driven by three uniformly sampled discrete actions (up force, no force, down force). Initial states are sampled on energy contours with radius in $[0.5, 1.8]$ following [28]. The dataset contains 50,000 sequences of 50 frames at $\Delta t_{\mathrm{train}} = 0.4$ (2.5 Hz sampling).

Evaluation. We evaluate the oscillator at $\Delta t_{\mathrm{eval}} \in \{0.05, 0.1, 0.15, 0.2, 0.3, 0.4, 0.5, 0.6, 0.8, 1.0, 1.5\}$. for each evaluation trajectory we sample a single initial state and a reference action sequence at the training step size, then render the same dynamical trajectory at every Δt_{eval} by integrating from the shared initial state at that step size, with the reference actions resampled so the per-action wall-clock impulses are preserved. The temporal horizon is the same across Δt_{eval} values; what varies is the number of integration observation steps that cover it. For the substepping experiments, each observation step is split into $N \in \{1, 2, 4\}$ integration substeps with the action impulse preserved.

Metrics. We evaluate prediction quality in pixel space over the full open-loop rollout, reporting mean absolute error (MAE), PSNR [17], SSIM [31], and the perceptual LPIPS distance [33] between predicted and ground-truth frames at every evaluation step size. These visual metrics primarily measure the differences in visual attributes between two images and, therefore, are not a perfect measure for trajectory error of physical dynamics. Since the physical state is unknown for the predicted trajectories in this setting, we rely on visual reconstructions and aggregated metrics across the full rollout horizon.

6 Failure Decomposition

When we evaluate the trained port-Hamiltonian model in the extrapolation regime, two visually distinct failure modes appear, each with a different cause and a different fix (Fig. 1).

6.1 Energy Magnitude Growth

With an unconstrained action map $G(a)$, predicted latents diverge in amplitude for step sizes beyond training, producing high-frequency artifacts in the decoded frames (see Fig. 1).

At large step sizes the numerical solver overshoots, advancing the state further per step than the true dynamics warrant. An unconstrained force map exploits this overshoot, injecting energy along the directions where the solver is least stable. Over multiple steps the injection compounds, producing the observed amplitude blow-up. From a spectral perspective, the action impulse has eigenvalues exceeding unity along certain phase-space directions at large Δt, and an unbounded G compounds directly into those directions.

Applying spectral normalization [21] to G bounds its norm to 1, capping per-step energy injection. Action energy then enters linearly over the rollout and the amplitude blow-up is eliminated across the full tested Δt range. We refer to this spectrally-normed variant as the *bounded* port-HGN and use it in all experiments that follow.

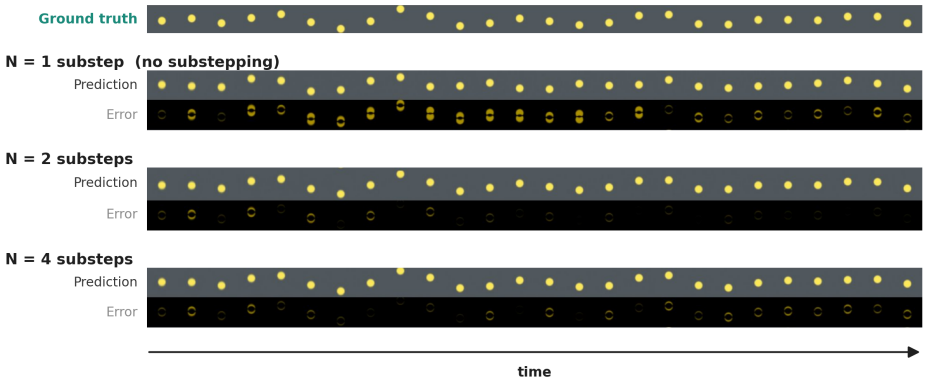

Fig. 3: Eval-time substepping recovers extrapolation on the oscillator. Top: ground-truth rollout. Each block below shows the spectrally-normed model's prediction (gray) and its error against the ground truth (black) at $N \in \{1, 2, 4\}$ integration substeps. At N=1 the prediction drifts out of phase and the error grows into bright double-lobed residuals. Refining the integration to N=2 and N=4 shrinks the error to faint traces, with the dynamics recovered once the effective step approaches the training scale.

6.2 Phase Drift Accumulation

With the action force constrained, a qualitatively different failure remains. Above $\Delta t = 0.4$, (the training step size), the prediction no longer diverges in amplitude; instead it drifts out of phase with the ground truth, and the per-frame error grows into the bright double-lobed residuals of a position offset (Fig. 3, N=1). The model does not explode; it loses temporal alignment.

The two candidate causes are a representational failure of the learned Hamiltonian beyond the training Δt regime, or global truncation error accumulating in the fixed-step integrator. Inference-time substepping targets exclusively the second explanation, since it refines the integration without altering the learned dynamics or the inferred initial state. Refining the integration progressively removes the drift: at N=2 substeps the error collapses to faint traces and at N=4 it is almost entirely gone (Fig. 3), the same recovery that appears across all four aggregate metrics (Fig. 4). The effective integration step at Δt=1.5 with N=4 is $1.5/4 \approx 0.375 \approx \Delta t_{\text{train}}$, and correct dynamics return when the integration step is brought back to the training scale. Because the learned dynamics are identical across these conditions, the recovery suggests that the failure is global truncation error rather than a deficiency of the learned Hamiltonian. In other words, when integrated at a step comparable to training, the same vector field reproduces correct dynamics at observation intervals well outside the training range.

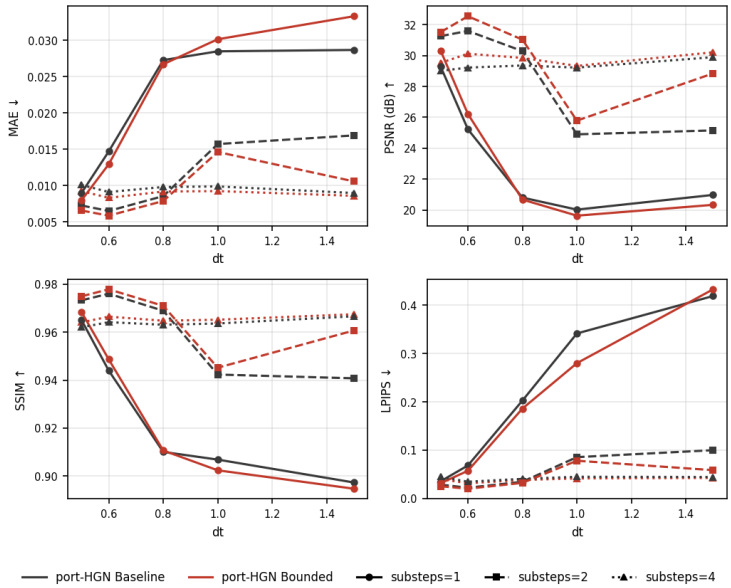

Fig. 4: Aggregate visual-quality metrics versus evaluation step size Δt_{eval}, restricted to step sizes beyond the training step (Δt_{train}=0.4), for the baseline port-HGN (black) and the spectrally-normed *bounded* port-HGN (red). Line style encodes the number of inference-time integration substeps (N=1 solid, N=2 dashed, N=4 dotted). Without substepping (N=1) both models degrade steeply as Δt_{eval} grows; increasing N restores quality across MAE, PSNR, SSIM, and LPIPS, with the N=4 curves remaining high and nearly flat in Δt_{eval}. At intermediate substepping (N=2) the bounded model recovers more than the baseline.

6.3 Recovering Both Directions

Taken together, the two fixes act on the two failure modes independently and combine into a single model that generalizes in both temporal directions. Figure 4 reports prediction quality across the extrapolation grid for the baseline and the spectrally-bounded port-HGN at $N \in \{1, 2, 4\}$ integration substeps. Without substepping (N=1) all four metrics degrade as Δt_{eval} grows, most sharply the perceptual LPIPS distance, while SSIM and MAE change least in absolute terms, as expected for a scene with a large static background. Substepping recovers every metric monotonically, and by N=4 the curves are nearly flat in Δt_{eval}, the effective integration step having returned to the training scale. The bounded model recovers more completely than the baseline at intermediate substepping (N=2), indicating that constraining the forcing and refining the integration address complementary defects rather than the same one.

Because spectral normalization constrains only the action-force map and substepping is engaged only when the target step exceeds the training step, the two fixes address the extrapolation failures specifically. Interpolation below the training step is provided by the symplectic integrator, which refines any sub-training step without retraining (Fig. 2), while subdividing large steps restores prediction above it (Fig. 3). A single port-Hamiltonian model trained at one frame rate therefore predicts at step sizes both finer and coarser than any it observed during training.

7 Discussion

For deployment of Hamiltonian video models at variable frame rates, our results suggest a practical recipe: apply spectral normalization to any learned forcing term, and substep the integrator at inference time when the target playback rate exceeds the training rate. The computational cost of substepping is proportional to N additional forward passes through the Hamiltonian network per observation step, and no retraining is required. The diagnosis attributes the two extrapolation failures to specific, individually removable causes: exponential magnitude growth to the unconstrained action-force map, and loss of temporal alignment to global truncation error in the fixed-step integrator.

8 Conclusion

A port-Hamiltonian video model grounds its predictions in a continuous-time energy function that is, in principle, queryable at any frame rate, and interpolation below the training step follows directly from its symplectic structure. In practice, however, this generalization breaks down once the rollout step grows beyond the training regime. We have shown that the breakdown is not a limitation of the learned representation but the product of two distinct numerical failure modes, namely latent magnitude growth driven by an unconstrained action-force map, and global truncation error accumulation from an under-resolved integrator. We provide a targeted fix for each, applying spectral normalization to the action-force map and substepping the integrator at inference time, and together they recover stable dynamics prediction at temporal resolutions well outside the training distribution, from a single model queried at rates both finer and coarser than the one it was trained on. These results recommend a practical recipe for enabling temporal generalization in continuous-time video generation.

References

1. Alonso, E., Jelley, A., Micheli, V., Kanervisto, A., Storkey, A.J., Pearce, T., Fleuret, F.: Diffusion for world modeling: Visual details matter in atari. Adv. Neural Inform. Process. Syst. (2024)

2. Assran, M., Duval, Q., Misra, I., Bojanowski, P., Vincent, P., Rabbat, M., LeCun, Y., Ballas, N.: Self-supervised learning from images with a joint-embedding predictive architecture. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) (2023)
3. Balestriero, R., LeCun, Y.: LeJEPA: Provable and scalable self-supervised learning without the heuristics. arXiv preprint arXiv:2511.08544 (2025)
4. Bar, A., Zhou, G., Tran, D., Darrell, T., LeCun, Y.: Navigation world models. 2025 IEEE Conf. Comput. Vis. Pattern Recog. pp. 15791–15801 (2024)
5. Brandstetter, J., Worrall, D.E., Welling, M.: Message passing neural PDE solvers. In: International Conference on Learning Representations (ICLR) (2022)
6. Chen, R.T.Q., Rubanova, Y., Bettencourt, J., Duvenaud, D.: Neural ordinary differential equations. In: Advances in Neural Information Processing Systems (NeurIPS) (2018)
7. Cranmer, M., Greydanus, S., Hoyer, S., Battaglia, P., Spergel, D., Ho, S.: Lagrangian neural networks. In: ICLR Workshop on Integration of Deep Neural Models and Differential Equations (2020)
8. Desai, S.A., Mattheakis, M., Sondak, D., Protopapas, P., Roberts, S.J.: Port-Hamiltonian neural networks for learning explicit time-dependent dynamical systems. Physical Review E **104**(3), 034312 (2021)
9. Greydanus, S., Dzamba, M., Yosinski, J.: Hamiltonian neural networks. In: Advances in Neural Information Processing Systems (NeurIPS) (2019)
10. Gumbsch, C., Sajid, N., Martius, G., Butz, M.V.: Learning hierarchical world models with adaptive temporal abstractions from discrete latent dynamics. In: International Conference on Learning Representations (ICLR) (2024)
11. Ha, D., Schmidhuber, J.: Recurrent world models facilitate policy evolution. In: Advances in Neural Information Processing Systems (NeurIPS) (2018)
12. Ha, D.R., Schmidhuber, J.: World models. ArXiv **abs/1803.10122** (2018)
13. Hafner, D., Lee, K.H., Fischer, I., Abbeel, P.: Deep hierarchical planning from pixels. In: Advances in Neural Information Processing Systems (NeurIPS) (2022)
14. Hafner, D., Lillicrap, T., Ba, J., Norouzi, M.: Dream to control: Learning behaviors by latent imagination. In: International Conference on Learning Representations (ICLR) (2020)
15. Hafner, D., Lillicrap, T.P., Norouzi, M., Ba, J.: Mastering atari with discrete world models. In: Int. Conf. Learn. Represent. (2021)
16. Hafner, D., Pasukonis, J., Ba, J., Lillicrap, T.: Mastering diverse control tasks through world models. Nature **640**, 647–653 (2025)
17. Horé, A., Ziou, D.: Image quality metrics: PSNR vs. SSIM. In: Proceedings of the 20th International Conference on Pattern Recognition (ICPR). pp. 2366–2369 (2010)
18. Kingma, D.P., Welling, M.: Auto-encoding variational Bayes. In: International Conference on Learning Representations (2014), arXiv:1312.6114
19. LeCun, Y.: A path towards autonomous machine intelligence. Technical report, OpenReview (2022), https://openreview.net/forum?id=BZ5a1r-kVsf
20. Maes, L., Le Lidec, Q., Scieur, D., LeCun, Y., Balestriero, R.: LeWorldModel: Stable end-to-end joint-embedding predictive architecture from pixels. arXiv preprint arXiv:2603.19312 (2026)
21. Miyato, T., Kataoka, T., Koyama, M., Yoshida, Y.: Spectral normalization for generative adversarial networks. In: International Conference on Learning Representations (2018)

22. Peng, X.B., Andrychowicz, M., Zaremba, W., Abbeel, P.: Sim-to-real transfer of robotic control with dynamics randomization. In: IEEE International Conference on Robotics and Automation (ICRA). pp. 3803–3810 (2018)
23. Po, R., Nitzan, Y., Zhang, R., Chen, B., Dao, T., Shechtman, E., Wetzstein, G., Huang, X.: Long-context state-space video world models. IEEE International Conference on Computer Vision **abs/2505.20171** (2025)
24. Rubanova, Y., Chen, R.T.Q., Duvenaud, D.: Latent ODEs for irregularly-sampled time series. In: Advances in Neural Information Processing Systems. vol. 32 (2019)
25. van der Schaft, A.J.: Port-Hamiltonian systems: Network modeling and control of nonlinear physical systems. In: Advanced Dynamics and Control of Structures and Machines, CISM Courses and Lectures, vol. 444, pp. 127–168. Springer (2004)
26. Sutton, R.S., Precup, D., Singh, S.: Between MDPs and semi-MDPs: A framework for temporal abstraction in reinforcement learning. Artificial Intelligence **112**(1–2), 181–211 (1999)
27. Tallec, C., Blier, L., Ollivier, Y.: Making deep Q-learning methods robust to time discretization. In: International Conference on Machine Learning (ICML). pp. 6096–6104 (2019)
28. Toth, P., Rezende, D.J., Jaegle, A., Racanière, S., Botev, A., Higgins, I.: Hamiltonian generative networks. In: International Conference on Learning Representations (ICLR) (2020)
29. Valevski, D., Leviathan, Y., Arar, M., Fruchter, S.: Diffusion models are real-time game engines. International Conference on Learning Representations (ICLR) **abs/2408.14837** (2024)
30. Verlet, L.: Computer "experiments" on classical fluids. i. thermodynamical properties of lennard-jones molecules. Physical Review **159**, 98–103 (1967), https://api.semanticscholar.org/CorpusID:124980807
31. Wang, Z., Bovik, A.C., Sheikh, H.R., Simoncelli, E.P.: Image quality assessment: From error visibility to structural similarity. IEEE Transactions on Image Processing **13**(4), 600–612 (2004)
32. Watter, M., Springenberg, J.T., Boedecker, J., Riedmiller, M.: Embed to control: A locally linear latent dynamics model for control from raw images. In: Advances in Neural Information Processing Systems (NeurIPS) (2015)
33. Zhang, R., Isola, P., Efros, A.A., Shechtman, E., Wang, O.: The unreasonable effectiveness of deep features as a perceptual metric. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR). pp. 586–595 (2018)
34. Zhang, W., Terver, B., Zholus, A., Chitnis, S., Sutaria, H., Assran, M., Balestriero, R., Bar, A., Bardes, A., LeCun, Y., Ballas, N.: Hierarchical planning with latent world models. In: arxiv (2026), https://api.semanticscholar.org/CorpusID:287122091
35. Zhou, G., Pan, H., LeCun, Y., Pinto, L.: DINO-WM: World models on pre-trained visual features enable zero-shot planning. 2025 Int. Conf. Machine Learning (2025)