LaWM: Least Action World Models for Long-Horizon Physical Consistency from Visual Observations

Qixin Xiao qxiaocs@umich.edu

Maani Ghaffari maanigj@umich.edu

University of Michigan

Abstract

Learning predictive world models from visual observations is a core problem in embodied AI, with applications to model-based reinforcement learning and robotic planning. Existing latent world models typically generate future states with unconstrained neural transition functions, while modern video generation systems often prioritize perceptual plausibility or introduce physical structure through auxiliary losses, external guidance, or separate dynamics modules. As a result, long-horizon rollouts can remain weakly grounded in the physical principles that govern real dynamics, leading to compounding error, energy drift, and physically inconsistent futures. We propose **Least Action World Models** (LaWM), a latent world-modeling framework that operationalizes the *Principle of Least Action* in learned visual latent space: future rollouts are governed by a learned Lagrangian action functional rather than produced only by an unconstrained transition predictor. Our main technical realization is a *latent variational integrator*: LaWM encodes observations into learned generalized coordinates, learns a latent discrete Lagrangian over consecutive latent states, constructs a discrete action functional, and advances prediction by solving the corresponding discrete integration condition. Thus, physical structure is not merely used to score, regularize, or constrain a completed trajectory; it defines the latent transition rule itself. Because the transition is induced by a discrete variational principle, LaWM provides a structure-preserving bias for long-horizon visual prediction. Across physics-clean synthetic dynamics and embodied robot interaction benchmarks, LaWM improves physical invariance, background consistency, motion smoothness, and appearance and geometric prediction metrics over video-generation and world-model baselines.

1 Introduction

Learning predictive world models from visual observations is a core problem in embodied AI, with applications to model-based reinforcement learning and robotic planning. A useful visual world model should do more than synthesize plausible future frames: it should produce rollouts that remain dynamically coherent over long horizons. This is challenging because images contain high-dimensional appearance variation, while the underlying evolution is often governed by lower-dimensional physical structure.

Most latent world models address this problem by encoding observations into compressed states and learning a transition function in latent space. This design has been effective for prediction and control, but the transition itself is usually an unconstrained neural predictor. Modern video generation models approach prediction from the synthesis side and can produce visually plausible motion, but *perceptual plausibility does not guarantee physical consistency.* In both cases, a rollout may look reasonable over short horizons while gradually violating the regularities that govern real dynamics, leading to compounding error, energy drift, unstable acceleration, or inconsistent geometry.

Preprint.

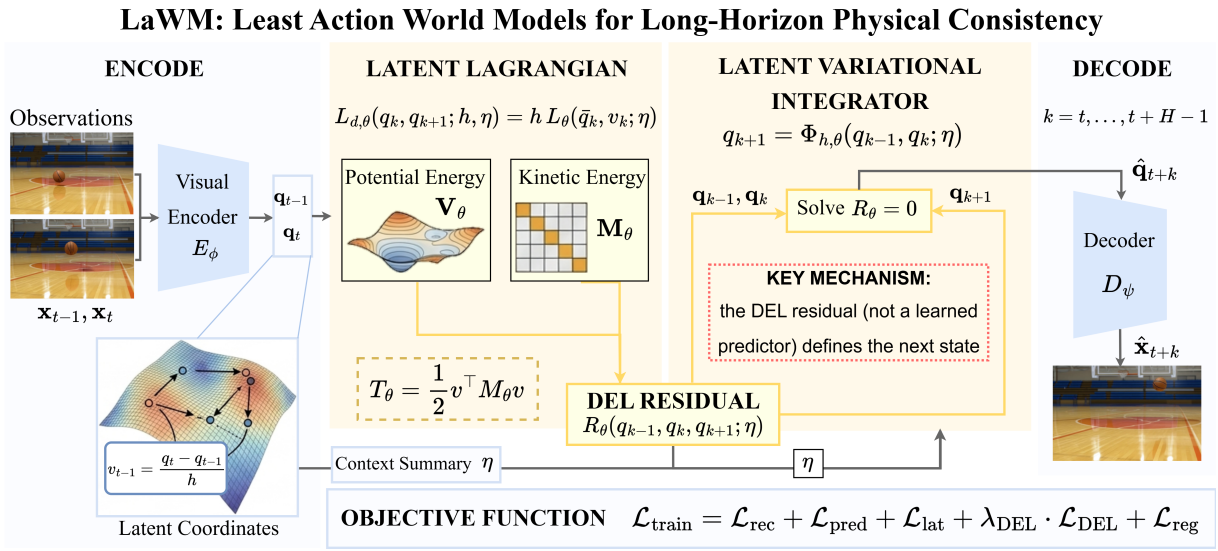


Figure 1: Overview of LaWM. Observations are encoded into latent coordinates, where a learned discrete Lagrangian defines a DEL residual. The next latent state is obtained by solving the DEL condition and is then decoded into future observations.

Physics-informed learning offers a natural path toward more reliable rollouts. Recent physics-aware video models improve physical plausibility through external guidance, auxiliary objectives, structured conditioning, learned dynamics modules, or simulation-rendering pipelines [13, 19, 21, 18]. These mechanisms are effective, but they usually place physical structure around the generative process rather than making the physical principle itself the latent transition rule. In other words, they can evaluate, guide, or correct prediction, but they do not directly change the central object of a world model: the rule by which one latent state advances to the next. For long-horizon world modeling, physical structure should not merely evaluate or correct prediction; it should help construct prediction.

We propose **Least Action World Models** (LaWM), which operationalizes the *Principle of Least Action* in learned visual latent space. Under this principle, a physically admissible trajectory is characterized by the stationarity of a Lagrangian action functional. We adapt this idea to world modeling by requiring future rollouts to be governed by a learned action functional, rather than produced only by an unconstrained neural transition. Here, "action" refers to the Lagrangian action functional, not a control command. This framing separates the high-level principle from its implementation. The learned action functional provides an organizing object for physical rollout, while the transition mechanism determines how strongly this structure is built into prediction.

Our main technical realization of this idea is a *latent variational integrator*. LaWM maps observations into learned generalized coordinates, learns a latent discrete Lagrangian over pairs of consecutive latent states, constructs a discrete action functional, and advances prediction by solving the corresponding discrete Euler–Lagrange (DEL) condition. The key distinction is that the action functional is not used only to score, regularize, or refine a completed trajectory. Instead, its stationarity condition defines the latent transition rule itself. In this sense, least action becomes the rollout mechanism of the world model.

This formulation is especially natural for video because observations and predictions are already discrete in time. Rather than learning a continuous-time dynamics model and then applying an arbitrary discretization, LaWM learns a discrete Lagrangian directly on encoded video states. Following discrete mechanics and variational integrators [16], the resulting transition is induced by a discrete variational principle, giving the latent rollout a structure-preserving bias for long-horizon prediction without assuming access to true physical coordinates.

We evaluate LaWM in two complementary regimes. The physics-clean synthetic setting tests whether the model preserves motion-specific physical quantities across canonical dynamics, including uniform motion, acceleration, parabolic motion, slope sliding, rotation, damped oscillation, scale variation, and deformation. The embodied robot interaction setting evaluates whether the same transition-

level physical structure improves prediction under clutter, occlusion, contact-rich motion, depth variation, and embodied scene evolution. Across both regimes, LaWM improves physical invariance, background consistency, motion smoothness, and appearance and geometric prediction metrics over relevant video-generation and world-model baselines.

Our main contributions are as follows:

1. We introduce **Least Action World Models**, a latent world-modeling framework that operationalizes the *Principle of Least Action* by governing future rollouts with a learned Lagrangian action functional rather than only an unconstrained neural transition.
2. We formulate a **latent discrete mechanics model** for visual prediction, where encoded observations serve as learned generalized coordinates and a learned latent discrete Lagrangian defines the action functional.
3. We derive a **latent variational integrator** from the discrete Euler–Lagrange condition, turning least action from a trajectory-level objective into the transition rule used for rollout.
4. We show that LaWM improves long-horizon physical consistency and embodied prediction across physics-clean dynamics and robot interaction benchmarks.

2 Related Work

2.1 Latent World Models and Video Generation

Latent world models learn compressed state representations in which prediction, planning, and control can be performed more efficiently than in pixel space. PlaNet learns latent dynamics from pixel observations and uses online planning in the learned latent space for model-based control [9]. Dreamer trains policies by backpropagating through imagined trajectories in a learned latent world model, establishing latent imagination as an effective control paradigm [8]. DreamerV3 scales world-model learning across diverse domains with a fixed algorithmic configuration, but still relies on learned predictive transitions rather than a physical variational transition rule [10]. Deep Koopman models learn nonlinear coordinates in which dynamics can be approximated by a linear evolution operator, offering another representation-space view of temporal prediction [15]. V-JEPA learns video representations by predicting future latent features from context, showing the value of representation-space prediction without pixel reconstruction [1].

Video generation models approach prediction from the synthesis side. VideoLDM extends latent diffusion to video by generating in compressed visual latents, reducing the compute and memory cost of high-resolution synthesis [2]. Sora frames large-scale video generation as a path toward world simulation, highlighting the possibility that generative video models may acquire implicit physical understanding at scale [5]. CogVideoX uses a 3D VAE and diffusion transformer to improve long-duration text-to-video generation in compressed video latents [20].

These works show the power of latent prediction and video synthesis, but most latent world models still use data-driven transition operators, while video generators mainly optimize perceptual or appearance-level plausibility. LaWM differs by treating encoded visual states as dynamics-aware latent coordinates and deriving the rollout from the Principle of Least Action.

2.2 Physics-Informed Dynamics and Physics-Guided Video

Physics-informed learning studies how physical structure can improve prediction, stability, and generalization. Hamiltonian Neural Networks learn a Hamiltonian function and use Hamiltonian dynamics to improve energy behavior over standard neural predictors [7]. Lagrangian Neural Networks learn a Lagrangian directly and do not require canonical coordinates, making them suitable when momenta are difficult to observe [6]. Physics-Informed Neural Networks enforce differential-equation constraints during training and show that known physical laws can serve as strong supervision for scientific learning [17]. Karniadakis et al. survey physics-informed machine learning as a broad paradigm for combining data with physical models and constraints [11].

Recent video-generation methods also inject physics into generative pipelines. PhysGen combines image-based physical understanding, rigid-body simulation, and generative video rendering to produce physically plausible image-to-video results [13]. NewtonGen introduces Neural Newtonian

Dynamics to predict Newtonian motion and guide text-to-video generation with controllable physical trajectories [21]. RoboScape jointly trains RGB video generation with temporal depth prediction and keypoint dynamics learning to improve geometric and physical consistency in embodied robot videos [18].

These methods demonstrate that physics improves visual prediction, but they often introduce physics through auxiliary losses, separate dynamics modules, simulation stages, or joint supervision tasks. LaWM targets a different layer: the transition rule itself. Instead of attaching physics around a generator or adding it only as a loss, LaWM places least action inside the latent world-model transition.

2.3 Discrete Mechanics and Variational Integrators

Discrete mechanics provides the theoretical foundation for LaWM. Marsden and West formulate discrete mechanics by replacing the continuous tangent bundle TQ with pairs of configurations $Q \times Q$, defining a discrete Lagrangian $L_d(q_k, q_{k+1})$, summing it into a discrete action, and deriving the discrete Euler–Lagrange equations from stationary action [16]. Marsden and West also show that the discrete Euler–Lagrange equations define a recursive discrete Lagrangian map, which is the variational integrator induced by the discrete action principle [16]. Hairer, Lubich, and Wanner study geometric numerical integration as a general framework for structure-preserving algorithms, including symplectic methods for Hamiltonian systems [14]. Leimkuhler and Reich present Hamiltonian simulation methods that emphasize symplectic structure and long-time numerical behavior in mechanical systems [12]. Bloch, Crouch, and Ratiu connect symmetric discrete optimal control with backpropagation and deep learning, showing another link between discrete geometric structure and learning-based optimization [4]. Bloch, Puiggalí Farré, and de Diego extend Euler–Poincare dynamics to metriplectic systems, which combine Hamiltonian structure with entropy production to model dissipative open systems [3].

LaWM brings this discrete-mechanics viewpoint into visual world modeling. Rather than learning a continuous-time physical law and retrofitting it to frame sequences, LaWM learns a latent discrete Lagrangian directly on pairs of encoded video states. The resulting discrete action functional yields a DEL condition, and the LaWM variational transition uses this condition to generate each next latent state. Thus, LaWM is not a video generator with physics attached; it is a world model whose latent transition is derived from the Principle of Least Action.

3 Method

We now formalize **Least Action World Models (LaWM)** as a latent variational transition model. Given an observation sequence, LaWM encodes each time step into a latent coordinate, learns a latent discrete Lagrangian on pairs of neighboring latent states, and uses the resulting discrete Euler–Lagrange (DEL) condition to define the next-state update. The resulting transition is a *latent variational integrator*: each predicted latent state is obtained by solving a local DEL equation rather than by applying an unconstrained neural transition.

We first define the latent prediction interface, then construct a latent discrete Lagrangian, form the corresponding discrete action, derive the DEL residual, and use this residual as a recursive rollout rule. Throughout this section, *action* refers to the Lagrangian action functional, not a robot control input. Since visual observations arrive at discrete time steps, we formulate the dynamics directly in discrete time. Following discrete mechanics, a pair of neighboring configurations plays the role of a discrete position–velocity state, and a discrete Lagrangian on such pairs induces a time-stepping map through stationary action [16].

3.1 Problem Setup and Latent World Model

We begin with the standard latent-world-model interface, because LaWM changes the rollout rule rather than the observation format. Let $y_{1:T}$ denote an observed sequence. For visual prediction, $y_t = x_t \in \mathbb{R}^{C\times H\times W}$ is an image frame. When generalized states are available, $y_t = s_t \in \mathbb{R}^{d_s}$ is a physical state. LaWM maps each observation to a latent coordinate

$$q_t = E_\phi(y_t), \qquad q_t \in \mathbb{R}^d. \tag{1}$$

The predicted output is obtained through a readout map

$$\hat{y}_t = D_\psi(q_t). \tag{2}$$

For state-supervised training, E_ϕ and D_ψ may be identity maps. For visual training, they are implemented as an encoder and decoder.

A standard latent world model can be abstracted as learning a direct transition predictor

$$q_{t+1} = f_\omega(q_t), \tag{3}$$

or a recurrent variant of the same idea. Such transitions can be effective over short horizons, but the update rule itself is not derived from a physical action principle. LaWM keeps the same latent prediction interface, but replaces the primary rollout rule with a second-order variational transition

$$\hat{q}_{k+1} = \Phi_{h,\theta}(\hat{q}_{k-1}, \hat{q}_k; \eta), \tag{4}$$

where h is the time step and η is a sequence-level latent physical context inferred from the observed frames:

$$\eta = g_\rho(q_{t-1}, q_t, q_t - q_{t-1}). \tag{5}$$

The context parameter η is learned end-to-end, is not a motion-category label, and is held fixed during the rollout horizon. The remaining subsections define the variational map $\Phi_{h,\theta}$.

3.2 Latent Discrete Lagrangian

To make the latent rollout governed by least action, we first define an action functional in latent space. In classical mechanics, this starts from a Lagrangian

$$L(q, \dot{q}) = T(q, \dot{q}) - V(q), \tag{6}$$

where T is kinetic energy and V is potential energy. In LaWM, the latent coordinate q_t plays the role of a learned generalized coordinate. Because videos are observed at discrete time steps, we do not learn a continuous-time equation and then discretize it afterward. Instead, we learn a *latent discrete Lagrangian*

$$L_{d,\theta}(q_k, q_{k+1}; h, \eta), \tag{7}$$

defined directly on pairs of consecutive latent coordinates.

For each pair (q_k, q_{k+1}), we define the midpoint coordinate and discrete velocity

$$\bar{q}_k = \frac{q_k + q_{k+1}}{2}, \qquad v_k = \frac{q_{k+1} - q_k}{h}. \tag{8}$$

We parameterize the latent Lagrangian as

$$L_\theta(q, v; \eta) = \frac{1}{2} v^\top M_\theta(q, \eta) v - V_\theta(q, \eta), \tag{9}$$

where V_θ is a scalar potential network and M_θ is a positive diagonal latent mass matrix. We use

$$M_\theta(q, \eta) = \mathrm{diag}(m_\theta(q, \eta)) + \epsilon I, \qquad m_\theta(q, \eta) > 0, \tag{10}$$

with m_θ parameterized by a positive activation. This ensures that the kinetic term is non-negative.

The midpoint discrete Lagrangian is then

$$L_{d,\theta}(q_k, q_{k+1}; h, \eta) = h L_\theta\left(\bar{q}_k, v_k; \eta\right). \tag{11}$$

This construction should be read as a learned latent mechanics model, not as a claim that the latent variables are exact physical coordinates. The role of $L_{d,\theta}$ is to provide the action functional from which the rollout rule will be derived.

3.3 Discrete Action and Latent Euler–Lagrange Equation

The learned discrete Lagrangian is not yet a transition rule. To turn it into one, we form the discrete action over a latent trajectory

$$Q_{t:t+H} = \{q_t, q_{t+1}, \ldots, q_{t+H}\}, \tag{12}$$

as

$$\mathcal{S}_{d,\theta}(Q_{t:t+H};\eta) = \sum_{k=t}^{t+H-1} L_{d,\theta}(q_k, q_{k+1}; h, \eta). \tag{13}$$

Under the discrete action principle, a trajectory is variationally admissible if $\mathcal{S}_{d,\theta}$ is stationary with respect to variations of the interior latent states, with endpoints fixed. Taking this variation gives the discrete Euler–Lagrange equation

$$D_2 L_{d,\theta}(q_{k-1}, q_k; h, \eta) + D_1 L_{d,\theta}(q_k, q_{k+1}; h, \eta) = 0, \tag{14}$$

where D_1 and D_2 denote derivatives with respect to the first and second arguments of $L_{d,\theta}$.

We define the DEL residual as

$$R_\theta(q_{k-1}, q_k, q_{k+1}; \eta) = D_2 L_{d,\theta}(q_{k-1}, q_k; h, \eta) + D_1 L_{d,\theta}(q_k, q_{k+1}; h, \eta). \tag{15}$$

Equation (15) is the local stationarity condition that connects the previous interval (q_{k-1}, q_k) to the next interval (q_k, q_{k+1}). This is the central bridge from principle to algorithm: the action functional is not merely a score assigned to a completed rollout; its stationarity condition provides the equation that will determine each next latent state.

3.4 Latent Variational Integrator

The DEL condition can be read as an implicit equation for the unknown next state q_{k+1}. Given two consecutive latent states q_{k-1} and q_k, the exact variational update would choose q_{k+1} such that

$$R_\theta(q_{k-1}, q_k, q_{k+1}; \eta) = 0. \tag{16}$$

In practice, we approximate this local root solve with a fixed number of differentiable solver iterations:

$$q_{k+1} = \mathrm{Solve}_N\left(q_{k-1}, q_k; \eta\right), \qquad R_\theta(q_{k-1}, q_k, q_{k+1}; \eta) \approx 0. \tag{17}$$

This defines the *latent variational integrator*

$$q_{k+1} = \Phi_{h,\theta}(q_{k-1}, q_k; \eta). \tag{18}$$

The order of the arguments emphasizes the second-order nature of the update: the discrete state is the pair (q_{k-1}, q_k), and the integrator advances it to (q_k, q_{k+1}).

We initialize the local solve by constant-velocity extrapolation:

$$q'^{(0)} = q_k + (q_k - q_{k-1}). \tag{19}$$

We instantiate Solve_N as an unrolled residual-correction procedure,

$$q'^{(i+1)} = q'^{(i)} - \alpha P_\theta(q_k, \eta) R_\theta(q_{k-1}, q_k, q'^{(i)}; \eta), \qquad i = 0, \ldots, N-1, \tag{20}$$

where $P_\theta(q_k, \eta)$ is a positive diagonal preconditioner derived from the learned mass. After N iterations, we set $q_{k+1} = q'^{(N)}$, and gradients are backpropagated through the unrolled solver. Appendix A.5.2 ablates N and shows that $N = 4$ already recovers stable PIS and energy behavior.

The important distinction is that Equation (20) is a local root solve for the next state in the DEL equation. It is not gradient descent over a completed trajectory. In a post-hoc refinement approach, a full rollout is first generated and then optimized afterward using a trajectory-level objective. In LaWM, by contrast, the accepted next latent state is produced by the DEL solve itself, so the variational principle defines the transition.

Equivalently, the DEL condition matches the incoming and outgoing discrete momenta at q_k, giving the update its standard variational-integrator interpretation. This is not an additional module; it is another way to read the same DEL-defined transition.

Starting from encoded context states

$$\hat{q}_{t-1} = q_{t-1}, \qquad \hat{q}_t = q_t, \tag{21}$$

future latent states are generated recursively:

$$\hat{q}_{k+1} = \Phi_{h,\theta}(\hat{q}_{k-1}, \hat{q}_k; \eta), \qquad k = t, \ldots, t+H-1. \tag{22}$$

The predicted output is obtained from the predicted latent coordinate:

$$\hat{s}_{t+k} = \hat{q}_{t+k} \quad \text{for state-space LaWM}, \qquad \hat{x}_{t+k} = D_\psi(\hat{q}_{t+k}) \quad \text{for visual LaWM}. \tag{23}$$

Under exact unforced DEL solves, this construction is a variational integrator in latent space. In the learned visual setting, we do not assume exact physical coordinates or exact physical correctness in pixel space; instead, we treat the variational structure as an inductive bias and evaluate it empirically through long-horizon physical consistency, DEL residuals, and energy-drift diagnostics.

3.5 Learning Objective

The learning objective trains the encoder, decoder, latent Lagrangian, context encoder, and finite DEL solver so that the variational rollout matches observed data. Importantly, all predicted states in the losses below are generated by the latent variational integrator in Equation (22); the losses train the rollout mechanism but do not replace it with post-hoc trajectory optimization.

We write the full objective as

$$\mathcal{L}_{\text{train}} = \mathcal{L}_{\text{sup}} + \lambda_{\text{DEL}}\mathcal{L}_{\text{DEL}} + \lambda_{\text{reg}}\mathcal{L}_{\text{reg}}. \tag{24}$$

When generalized states are available during training, the supervised rollout loss is a weighted trajectory error,

$$\mathcal{L}_{\text{sup}} = \mathcal{L}_{\text{traj}} = \sum_{k=0}^{H} \|w \odot (\hat{s}_{t+k} - s_{t+k})\|_2^2 . \tag{25}$$

When training directly from visual observations, we first decode the reconstructed context frame and the predicted future frame as

$$\hat{x}_t^{\text{rec}} = D_\psi(E_\phi(x_t)), \qquad \hat{x}_{t+k} = D_\psi(\hat{q}_{t+k}). \tag{26}$$

The supervised visual term becomes

$$\mathcal{L}_{\text{sup}} = \lambda_{\text{rec}}\mathcal{L}_{\text{rec}} + \lambda_{\text{pred}}\mathcal{L}_{\text{pred}} + \lambda_{\text{lat}}\mathcal{L}_{\text{lat}}, \tag{27}$$

where

$$\mathcal{L}_{\text{rec}} = \sum_t \|\hat{x}_t^{\text{rec}} - x_t\|_2^2 , \tag{28}$$

$$\mathcal{L}_{\text{pred}} = \sum_{k=1}^{H} \|\hat{x}_{t+k} - x_{t+k}\|_2^2 , \tag{29}$$

$$\mathcal{L}_{\text{lat}} = \sum_{k=1}^{H} \|\hat{q}_{t+k} - \text{sg}(E_\phi(x_{t+k}))\|_2^2 . \tag{30}$$

Because Solve_N uses a finite number of residual-correction steps, the generated trajectory can have a nonzero DEL residual. We therefore include

$$\mathcal{L}_{\text{DEL}} = \sum_{k=1}^{H-1} \|R_\theta(\hat{q}_{t+k-1}, \hat{q}_{t+k}, \hat{q}_{t+k+1}; \eta)\|_2^2 . \tag{31}$$

This term stabilizes the approximate finite solve and provides a diagnostic for variational consistency. It is not the primary rollout mechanism: the rollout is still generated recursively by the DEL-defined transition in Equation (22).

Finally, $\mathcal{L}_{\text{reg}}$ contains standard parameter regularization and a mass-conditioning penalty $\sum_k \|\log m_\theta(\hat{q}_k, \eta)\|_2^2$. At inference time, LaWM infers η, recursively applies the latent variational integrator, and maps the predicted latent trajectory back to the observation space.

4 Experiments

4.1 Experimental Setup

We evaluate LaWM in two complementary regimes: physics-clean video prediction and embodied robot interaction prediction. The physics-clean benchmark follows the NewtonGen-style protocol [21] and covers canonical dynamics including uniform motion, acceleration, deceleration, parabolic motion, slope sliding, circular motion, rotation, damped oscillation, scale variation, and deformation. This setting tests whether a model preserves motion-specific physical regularities during long-horizon rollout. The embodied benchmark follows a RoboScape-style robot interaction setting [18] with RGB observations, depth, robot states, and embodied scene evolution, testing whether the same variational transition remains useful under clutter, occlusion, contact, and geometric complexity. Full

dataset construction, preprocessing, training details, prediction horizons, rendering protocol, metric definitions, and compute resources are provided in Appendix A.1, Appendix A.2, and Appendix A.4.

For physics-clean dynamics, we compare against strong video-generation and physics-oriented baselines, including Sora, Veo3, CogVideoX-5B, Wan2.2, PhyT2V, and NewtonGen. We report Physical Invariance Score (PIS), Background Consistency (BC), and Motion Smoothness (MS). For embodied prediction, we report appearance, depth, and action-conditioned metrics, including LPIPS, PSNR, AbsRel, δ_1, δ_2, and APSNR. All reported LaWM results use the same DEL-defined variational transition during training and inference.

4.2 Main Results

For closed-source or large-scale video generation baselines, we use the closest available reported NewtonGen-style benchmark results when available, and evaluate LaWM under the same metric definitions following the benchmark protocol.

Table 1: Physics-clean video prediction results across all motion types. All metrics in this table are higher is better. Values are reported as mean(std) when available. Green indicates the best learned method, and light green indicates the second-best learned method.

Motion Type	Metric	Reference	Sora	Veo3	CogVideoX-5B	Wan2.2	PhyT2V	NewtonGen	LaWM
Uniform Motion	PIS-v_x	0.9972	0.6548(0.022)	0.9784(0.006)	0.5392(0.007)	0.6395(0.029)	0.5349(0.014)	0.9830(0.005)	**0.9938(0.0045)**
Uniform Motion	BC	1	0.9573(0.003)	0.9491(0.024)	0.9534(0.018)	0.9683(0.027)	0.9612(0.015)	0.9694(0.020)	**0.9930(0.0021)**
Uniform Motion	MS	1	0.9926(0.003)	0.9953(0.001)	0.9905(0.005)	0.9939(0.003)	0.9876(0.015)	0.9962(0.003)	**0.9993(0.0001)**
Acceleration	PIS-a_x	0.8489	0.3437(0.355)	0.6187(0.308)	0.5458(0.038)	0.3077(0.261)	0.5033(0.011)	0.6568(0.013)	**0.8964(0.0275)**
Acceleration	BC	1	0.9495(0.011)	0.9373(0.015)	0.9518(0.037)	0.9695(0.018)	0.9636(0.021)	0.9748(0.012)	**0.9960(0.0030)**
Acceleration	MS	1	0.9852(0.011)	0.9909(0.004)	0.9876(0.008)	0.9908(0.005)	0.9822(0.010)	0.9918(0.009)	**0.9995(0.0001)**
Deceleration	PIS-a_x	0.8872	0.6162(0.072)	0.6173(0.102)	0.4988(0.014)	0.4705(0.328)	0.5167(0.023)	0.6891(0.007)	**0.8701(0.0448)**
Deceleration	BC	1	0.9494(0.026)	0.9295(0.039)	0.9623(0.017)	0.9721(0.012)	0.9622(0.012)	0.9744(0.012)	**0.9923(0.0025)**
Deceleration	MS	1	0.9883(0.006)	0.9933(0.003)	0.9787(0.024)	0.9903(0.007)	0.9814(0.014)	0.9947(0.005)	**0.9993(0.0001)**
Parabolic Motion	PIS-v_x	0.9988	0.9095(0.014)	0.9042(0.012)	0.7392(0.007)	0.7747(0.126)	0.6370(0.199)	0.9803(0.002)	**0.9961(0.0030)**
Parabolic Motion	PIS-a_y	0.9487	0.5723(0.266)	0.7662(0.139)	0.4230(0.028)	0.5571(0.953)	0.3567(0.799)	0.8189(0.014)	**0.9443(0.0170)**
Parabolic Motion	BC	1	0.9486(0.023)	0.9514(0.023)	0.9330(0.030)	0.9602(0.028)	0.9436(0.046)	0.9693(0.014)	**0.9882(0.0049)**
Parabolic Motion	MS	1	0.9915(0.004)	0.9948(0.002)	0.9856(0.009)	0.9903(0.007)	0.9844(0.011)	0.9967(0.001)	**0.9994(0.0000)**
3D Motion	PIS-Δl	0.7388	0.5013(0.005)	0.5932(0.005)	0.3026(0.005)	0.4583(0.005)	0.2911(0.007)	0.6472(0.005)	**0.9137(0.2333)**
3D Motion	PIS-v_y	0.9986	0.8481(0.008)	0.8913(0.008)	0.6690(0.003)	0.8384(0.018)	0.6510(0.002)	0.9371(0.007)	**0.9850(0.0081)**
3D Motion	BC	1	0.9426(0.017)	0.9410(0.022)	0.9620(0.018)	0.9772(0.008)	0.9629(0.016)	0.9672(0.018)	**0.9914(0.0055)**
3D Motion	MS	1	0.9934(0.003)	0.9944(0.003)	0.9945(0.003)	0.9943(0.002)	0.9888(0.012)	0.9954(0.005)	**0.9995(0.0001)**
Slope Sliding	PIS-a_x	0.8741	0.4931(0.153)	0.6081(0.157)	0.3533(0.160)	0.3108(0.421)	0.3570(0.354)	0.6312(0.041)	**0.8524(0.0383)**
Slope Sliding	PIS-a_y	0.9148	0.4616(0.212)	0.3815(0.092)	0.4731(0.028)	0.3967(0.744)	0.4297(0.569)	0.5840(0.043)	**0.8692(0.0560)**
Slope Sliding	BC	1	0.9667(0.013)	0.9631(0.016)	0.9556(0.024)	0.9653(0.017)	0.9568(0.022)	0.9787(0.010)	**0.9952(0.0021)**
Slope Sliding	MS	1	0.9919(0.005)	0.9958(0.002)	0.9903(0.006)	0.9912(0.005)	0.9829(0.014)	0.9971(0.001)	**0.9994(0.0001)**
Circular Motion	PIS-ω	0.9933	0.8393(0.010)	0.8932(0.007)	0.7726(0.026)	0.4677(0.006)	0.6391(0.322)	**0.9788(0.018)**	0.9562(0.0346)
Circular Motion	BC	1	0.9684(0.012)	0.9711(0.010)	0.9842(0.013)	0.9745(0.016)	0.9677(0.027)	0.9812(0.007)	**0.9906(0.0026)**
Circular Motion	MS	1	0.9949(0.001)	0.9960(0.001)	0.9979(0.001)	0.9949(0.004)	0.9974(0.002)	0.9980(0.001)	**0.9995(0.0001)**
Rotation	PIS-ω	0.9836	0.4267(0.099)	0.5285(0.436)	0.6596(0.023)	0.3425(0.172)	0.7842(0.304)	0.8838(0.038)	**0.9436(0.0197)**
Rotation	BC	1	0.9543(0.030)	0.9650(0.018)	0.9397(0.025)	0.9620(0.010)	0.9375(0.028)	0.9700(0.008)	**0.9850(0.0052)**
Rotation	MS	1	0.9900(0.006)	0.9942(0.003)	0.9795(0.028)	0.9909(0.007)	0.9878(0.006)	0.9958(0.002)	**0.9996(0.0000)**
Para. w/ Rotation	PIS-v_x	0.9990	0.5797(0.150)	0.7029(0.197)	0.6488(0.031)	0.6558(0.175)	0.7689(0.039)	0.9446(0.008)	**0.9943(0.0040)**
Para. w/ Rotation	PIS-a_y	0.9657	0.4903(2.581)	0.5603(1.012)	0.2614(0.127)	0.4331(1.982)	0.2879(0.046)	0.5614(0.028)	**0.9267(0.0290)**
Para. w/ Rotation	PIS-ω	0.9829	0.6522(0.556)	0.9019(0.119)	0.3380(0.199)	0.3474(0.334)	0.4119(0.105)	0.9289(0.029)	**0.9645(0.0147)**
Para. w/ Rotation	BC	1	0.9532(0.016)	0.9583(0.018)	0.9567(0.018)	0.9617(0.027)	0.9675(0.020)	0.9786(0.009)	**0.9869(0.0057)**
Para. w/ Rotation	MS	1	0.9889(0.005)	0.9952(0.001)	0.9841(0.018)	0.9908(0.005)	0.9921(0.003)	0.9969(0.001)	**0.9995(0.0000)**
Damped Oscillation	PIS-a_y	0.9402	0.4418(0.364)	0.3516(0.482)	0.3083(0.055)	0.3494(0.395)	0.2841(0.042)	0.5240(0.017)	**0.8153(0.1076)**
Damped Oscillation	BC	1	0.9738(0.013)	0.9666(0.011)	0.9699(0.007)	0.9715(0.014)	0.9708(0.010)	0.9743(0.012)	**0.9935(0.0026)**
Damped Oscillation	MS	1	0.9909(0.014)	0.9958(0.001)	0.9853(0.005)	0.9919(0.009)	0.9867(0.003)	0.9968(0.001)	**0.9990(0.0000)**
Size Changing	PIS-Δr	0.8501	0.2840(0.010)	0.4167(0.006)	0.5774(0.007)	0.1972(0.022)	0.4010(0.011)	0.6362(0.010)	**0.8263(0.3048)**
Size Changing	BC	1	0.9507(0.025)	0.9548(0.017)	0.9636(0.019)	0.9735(0.018)	0.9666(0.022)	0.9669(0.015)	**0.9928(0.0037)**
Size Changing	MS	1	0.9916(0.003)	0.9955(0.002)	0.9926(0.007)	0.9889(0.008)	0.9925(0.004)	0.9955(0.002)	**0.9994(0.0001)**
Deformation	PIS-Δl	0.9247	0.3626(0.004)	0.3466(0.017)	0.3550(0.002)	0.3515(0.043)	0.3601(0.003)	0.5492(0.005)	**0.6141(0.3621)**
Deformation	BC	1	0.9553(0.039)	0.9058(0.052)	0.9462(0.018)	0.9347(0.042)	0.9211(0.010)	0.9475(0.025)	**0.9911(0.0034)**
Deformation	MS	1	0.9941(0.004)	0.9940(0.006)	0.9935(0.009)	0.9903(0.009)	0.9867(0.001)	0.9957(0.001)	**0.9994(0.0000)**

Table 1 reports physics-clean video prediction results across canonical dynamics. LaWM achieves the best or second-best learned-model performance on nearly all reported metrics. The most important gains appear on PIS, which directly measures whether the predicted rollout preserves the motion-specific physical quantity. Compared with NewtonGen, the strongest physics-oriented baseline, LaWM improves physical invariance across acceleration, deceleration, slope sliding, parabolic

motion, rotation, scale variation, and coupled translational–rotational dynamics. These results support the central claim that using the discrete Euler–Lagrange condition as the rollout rule improves physical consistency, rather than only making predictions visually smoother.

4.2.1 Embodied Interaction and Geometric Prediction

Table 2 reports results on RoboScape-style embodied video prediction. LaWM improves over RoboScape on all reported appearance, geometry, and action-conditioned metrics: LPIPS improves from 0.1259 to 0.1138, PSNR from 21.8533 to 22.4176, AbsRel from 0.3600 to 0.3284, and APSNR from 3.3435 to 3.6128. These gains suggest that the variational latent rollout improves not only visual prediction quality, but also depth consistency and control-conditioned scene evolution.

Table 2: Results on RoboScape-style embodied video prediction. LPIPS and AbsRel are lower is better; all other metrics are higher is better.

Method	**LPIPS** $\downarrow$	**PSNR** $\uparrow$	**AbsRel** $\downarrow$	$\boldsymbol{\delta_1}$ $\uparrow$	$\boldsymbol{\delta_2}$ $\uparrow$	**APSNR** $\uparrow$
IRASim	0.6674	11.5698	0.6252	0.5013	0.7020	0.0269
iVideoGPT	0.4963	16.1236	0.7586	0.3480	0.5795	0.1144
Genie	0.1683	19.7571	0.4425	0.5435	0.7736	1.9871
CogVideoX	0.2180	17.5222	0.5243	0.6046	0.7599	–
RoboScape	0.1259	21.8533	0.3600	0.6214	0.8307	3.3435
LaWM	**0.1138**	**22.4176**	**0.3284**	**0.6492**	**0.8476**	**3.6128**

The embodied setting remains challenging because contact, actuation, occlusion, and dissipation introduce effects that are not fully captured by the current unforced formulation.

4.3 Ablation Study

We summarize the main ablations here and provide full tables and diagnostics in Appendix. The central comparison is against a gradient-based trajectory refinement baseline that uses the learned action only to optimize a completed trajectory, whereas LaWM uses the discrete Euler–Lagrange condition as the recursive transition rule.

As shown in Table 4, LaWM improves motion-balanced mPIS from 0.756 to 0.888 and row-wise mPIS from 0.742 to 0.904, winning $14/17$ PIS quantities, all 12 MS metrics, and $9/12$ BC metrics. The few cases where refinement is stronger mainly involve damped oscillation, size change, and deformation, which are expected limitations of the current unforced variational formulation. Additional ablations show that removing the variational transition, DEL loss, context, or learnable mass degrades long-horizon consistency, and long-horizon audits confirm that the advantage persists beyond the main prediction horizon.

5 Conclusion

We introduced **Least Action World Models (LaWM)**, a latent world-modeling framework that operationalizes the Principle of Least Action in learned visual latent space. Its main technical realization is a latent variational integrator: LaWM learns a latent discrete Lagrangian, constructs the corresponding discrete action, and uses the discrete Euler–Lagrange condition to define the latent transition rule. Rather than using physics as auxiliary regularization or post-hoc guidance, LaWM makes the action principle the mechanism by which latent rollouts are generated.

Across physics-clean dynamics and embodied robot interaction benchmarks, LaWM improves physical invariance, background consistency, motion smoothness, and appearance and geometric prediction metrics over relevant video-generation and world-model baselines. These results support transition-level physical structure as a path toward more stable and physically grounded visual world models. Future work will extend this formulation to forced and constrained discrete variational settings, incorporating controls, dissipation, contact, deformation, and manifold constraints.

References

[1] Adrien Bardes, Quentin Garrido, Jean Ponce, Xinlei Chen, Michael Rabbat, Yann LeCun, Mahmoud Assran, and Nicolas Ballas. Revisiting feature prediction for learning visual representations from video. *arXiv preprint arXiv:2404.08471*, 2024.

[2] Andreas Blattmann, Robin Rombach, Huan Ling, Tim Dockhorn, Seung Wook Kim, Sanja Fidler, and Karsten Kreis. Align your latents: High-resolution video synthesis with latent diffusion models. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 22563–22575, 2023.

[3] Anthony Bloch, Marta Farré Puiggalí, and David Martín de Diego. Metriplectic euler-poincar\'e equations: smooth and discrete dynamics. *arXiv preprint arXiv:2401.05220*, 2024.

[4] Anthony M Bloch, Peter E Crouch, and Tudor S Ratiu. Symmetric discrete optimal control and deep learning. *arXiv preprint arXiv:2404.06556*, 2024.

[5] Tim Brooks, Bill Peebles, Connor Holmes, Will DePue, Yufei Guo, Leo Jing, David Schnurr, Joe Taylor, Troy Luhman, Eric Luhman, et al. Video generation models as world simulators. *OpenAI Blog*, 1(8):1, 2024.

[6] Miles Cranmer, Sam Greydanus, Stephan Hoyer, Peter Battaglia, David Spergel, and Shirley Ho. Lagrangian neural networks. *arXiv preprint arXiv:2003.04630*, 2020.

[7] Samuel Greydanus, Misko Dzamba, and Jason Yosinski. Hamiltonian neural networks. *Advances in neural information processing systems*, 32, 2019.

[8] Danijar Hafner, Timothy Lillicrap, Jimmy Ba, and Mohammad Norouzi. Dream to control: Learning behaviors by latent imagination. *arXiv preprint arXiv:1912.01603*, 2019.

[9] Danijar Hafner, Timothy Lillicrap, Ian Fischer, Ruben Villegas, David Ha, Honglak Lee, and James Davidson. Learning latent dynamics for planning from pixels. In *International conference on machine learning*, pages 2555–2565. PMLR, 2019.

[10] Danijar Hafner, Jurgis Pasukonis, Jimmy Ba, and Timothy Lillicrap. Mastering diverse domains through world models. *arXiv preprint arXiv:2301.04104*, 2023.

[11] George Em Karniadakis, Ioannis G Kevrekidis, Lu Lu, Paris Perdikaris, Sifan Wang, and Liu Yang. Physics-informed machine learning. *Nature Reviews Physics*, 3(6):422–440, 2021.

[12] Benedict Leimkuhler and Sebastian Reich. *Simulating hamiltonian dynamics*. Cambridge university press, 2004.

[13] Shaowei Liu, Zhongzheng Ren, Saurabh Gupta, and Shenlong Wang. Physgen: Rigid-body physics-grounded image-to-video generation. In *European Conference on Computer Vision*, pages 360–378. Springer, 2024.

[14] Christian Lubich and Gerhard Wanner. *Geometric Numerical Integration: Structure-Preserving Algorithms for Ordinary Differential Equations*. Springer, 2006.

[15] Bethany Lusch, J Nathan Kutz, and Steven L Brunton. Deep learning for universal linear embeddings of nonlinear dynamics. *Nature communications*, 9(1):4950, 2018.

[16] Jerrold E Marsden and Matthew West. Discrete mechanics and variational integrators. *Acta numerica*, 10:357–514, 2001.

[17] Maziar Raissi, Paris Perdikaris, and George E Karniadakis. Physics-informed neural networks: A deep learning framework for solving forward and inverse problems involving nonlinear partial differential equations. *Journal of Computational physics*, 378:686–707, 2019.

[18] Yu Shang, Xin Zhang, Yinzhou Tang, Lei Jin, Chen Gao, Wei Wu, and Yong Li. Roboscape: Physics-informed embodied world model. *arXiv preprint arXiv:2506.23135*, 2025.

[19] Qiyao Xue, Xiangyu Yin, Boyuan Yang, and Wei Gao. Phyt2v: Llm-guided iterative self-refinement for physics-grounded text-to-video generation. In *Proceedings of the Computer Vision and Pattern Recognition Conference*, pages 18826–18836, 2025.

[20] Zhuoyi Yang, Jiayan Teng, Wendi Zheng, Ming Ding, Shiyu Huang, Jiazheng Xu, Yuanming Yang, Wenyi Hong, Xiaohan Zhang, Guanyu Feng, et al. Cogvideox: Text-to-video diffusion models with an expert transformer. *arXiv preprint arXiv:2408.06072*, 2024.

[21] Yu Yuan, Xijun Wang, Tharindu Wickremasinghe, Zeeshan Nadir, Bole Ma, and Stanley H Chan. Newtongen: Physics-consistent and controllable text-to-video generation via neural newtonian dynamics. *arXiv preprint arXiv:2509.21309*, 2025.

A Additional Experimental Details

A.1 Datasets

Physics-clean dynamics. The physics-clean benchmark follows the NewtonGen-style setting and covers a broad range of canonical dynamical patterns, including uniform motion, uniformly accelerated motion, decelerated motion, projectile motion, 3D motion effects, inclined-plane sliding, circular motion, rotation, projectile motion with self-rotation, damped oscillation, scale variation, and deformation. These motion types cover common physical structures in everyday scenes and provide a controlled testbed for evaluating whether a world model can capture different dynamical mechanisms.

For each motion type, videos are generated by a physics-based simulator with controllable parameters, including initial position, initial velocity, pose angle, angular velocity, world scale, friction coefficient, damping coefficient, rotation axis, object size, and object shape. When simulator state annotations are available, we retain physical quantities such as position, velocity, angle, angular velocity, and scale as auxiliary supervision and evaluation signals. Otherwise, the model is trained from videos only, and physical trajectories are estimated from predicted frames during evaluation. Following the NewtonGen protocol, most videos have a duration of approximately 1–2 seconds, with object velocities mainly ranging from 0 to 15 m/s.

We split the dataset into training, validation, and test sets. To evaluate generalization, the test split uses initial conditions and environment parameters that are distributionally separated from the training split. For each video, the model receives the first T_{ctx} frames as context and predicts the following T_{pred} frames. We report both aggregate performance and per-motion results to analyze how different dynamical structures affect model behavior.

RoboScape embodied videos. Beyond clean synthetic dynamics, we evaluate on a robot manipulation video setting following the RoboScape construction protocol. RoboScape is built from AgiBotWorld-Beta and contains multimodal embodied interaction data, including RGB sequences, depth sequences, robot action sequences, and robot state trajectories. This setting is substantially more challenging than synthetic physics-clean videos because the model must handle visual clutter, occlusion, contact-rich motion, and embodied scene evolution.

RoboScape constructs physics-aware training signals through automatically extracted visual priors. Temporal depth information is obtained from Video Depth Anything, and keypoint motion trajectories are extracted with SpatialTracker. The dataset is further processed by shot boundary detection, action semantic segmentation, keyframe and clip quality filtering, and regrouping based on action difficulty and scene type. The final benchmark contains approximately $50{,}000$ video segments, covering 147 tasks and 72 skills. Videos are divided into short clips of 16 frames sampled at 2 Hz, yielding approximately 6.5 million training clips. During inference, the first frame is used as the conditioning input, and the model autoregressively predicts the remaining 15 frames.

A.2 Implementation Details

Given an observed video sequence $\{x_1, \ldots, x_T\}$, the visual encoder maps each frame into a latent coordinate

$$q_t = E_\phi(x_t). \tag{32}$$

The decoder D_ψ reconstructs observations from the latent coordinates, and the context encoder estimates the sequence-level parameter

$$\eta = g_\rho(q_{t-1}, q_t, q_t - q_{t-1}). \tag{33}$$

The parameter η is learned end-to-end and is held fixed during each rollout.

The latent discrete Lagrangian is parameterized by the learned mass matrix $M_\theta(q, \eta)$ and potential network $V_\theta(q, \eta)$ described in Section 3.2. Future latent states are generated recursively by the LaWM variational transition. Each implicit DEL solve is initialized by constant-velocity extrapolation,

$$q'^{(0)} = q_k + (q_k - q_{k-1}), \tag{34}$$

and computed with a fixed number of differentiable solver iterations. The same rollout procedure is used during training and inference.

The training objective follows Section 3.5:

$$\mathcal{L}_{\text{train}} = \lambda_{\text{rec}}\mathcal{L}_{\text{rec}} + \lambda_{\text{pred}}\mathcal{L}_{\text{pred}} + \lambda_{\text{lat}}\mathcal{L}_{\text{lat}} + \lambda_{\text{DEL}}\mathcal{L}_{\text{DEL}} + \lambda_{\text{reg}}\mathcal{L}_{\text{reg}}. \tag{35}$$

Here, $\mathcal{L}_{\text{rec}}$ anchors the latent coordinate to visual observations, $\mathcal{L}_{\text{pred}}$ trains decoded future-frame prediction, $\mathcal{L}_{\text{lat}}$ aligns the variational rollout with stop-gradient future encodings, and $\mathcal{L}_{\text{DEL}}$ stabilizes approximate finite-iteration solves. There is no separate trajectory-refinement stage in the reported LaWM model.

For the physics-clean dataset, we use AdamW with learning rate 10^{-4}, cosine learning-rate decay, and batch size 64. Following the NewtonGen-style protocol, each motion category uses the same train-test split, input context length, prediction horizon, and image resolution as the reported baselines. Training a single motion category takes approximately two hours on one NVIDIA A100 80GB GPU.

For the RoboScape-style embodied setting, we follow the public clip-based training protocol and divide videos into 16-frame clips sampled at 2 Hz. The model predicts future frames autoregressively from the observed visual context. When robot control inputs, depth signals, or keypoint trajectories are available under the benchmark protocol, they are used as conditioning or auxiliary supervision signals for embodied prediction. The term "action" in LaWM continues to denote the Lagrangian action functional. For the RoboScape-style setting, the training configuration uses 4 NVIDIA A100 80GB GPUs, and a full training run on the evaluated split takes approximately 180–220 hours of wall-clock time.

A.3 Video Generation Details for Physics-Clean Visualizations

We provide photorealistic qualitative visualizations for the twelve physics-clean motion types to make each physical category visually interpretable. These visualizations are used only for illustration and are not used for training, metric computation, or model selection. Each visualization shows five temporally ordered frames from one selected sequence.

For most motion types, we follow a NewtonGen's video generation pipeline. Given a LaWM rollout state sequence, we first convert the predicted object trajectory, pose, and scale variables into object masks and dense optical-flow fields. The optical-flow fields specify how the target object should move across frames. We then use the NewtonGen noise-warping procedure to transform these motion fields into motion-guided latent noise, which is used by a text-to-video diffusion model to generate a photorealistic sequence. In our implementation, the video generator is CogVideoX-5B with Go-with-the-Flow LoRA. The text prompt specifies the visual scene, object category, lighting, and background appearance, while the temporal motion is controlled by the LaWM-derived optical-flow guidance.

The twelve qualitative strips below correspond to uniform motion, acceleration, deceleration, parabolic motion, 3D movement, slope sliding, circular motion, rotation, parabolic motion with rotation, damped oscillation, size changing, and deformation.

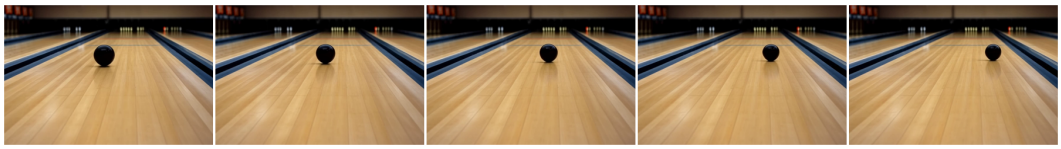


Figure 2: Uniform Motion. The object translates with approximately constant velocity.

Figure 3: Acceleration. The object moves with increasing displacement over time.

A.4 Evaluation Metrics

Visual prediction quality. We report standard future-frame prediction metrics, including PSNR, SSIM, and LPIPS. PSNR measures pixel-level reconstruction accuracy, SSIM measures structural

Figure 4: Deceleration. The object moves with decreasing displacement over time.

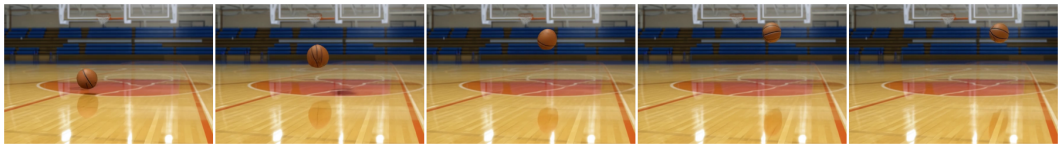


Figure 5: Parabolic Motion. The object follows a gravity-driven projectile trajectory.

Figure 6: 3D Movement. The object exhibits perspective-induced apparent scale and position change.

Figure 7: Slope Sliding. The object slides along an inclined surface under gravity.

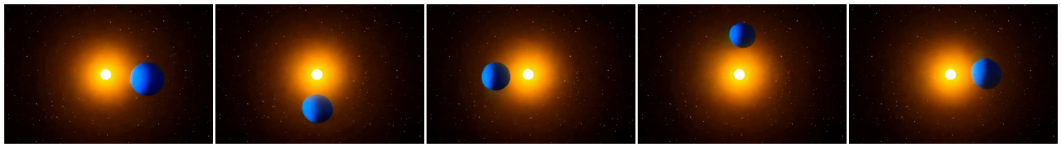


Figure 8: Circular Motion. The object follows an orbit-like circular trajectory around a central point.

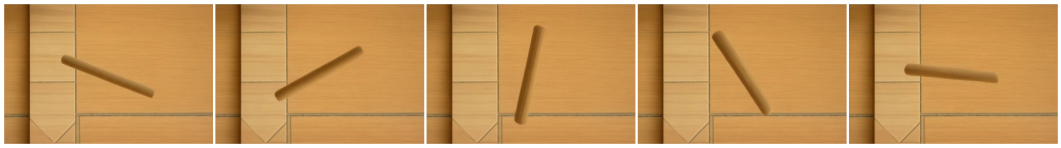


Figure 9: Rotation. The object rotates around its own center while remaining spatially localized.

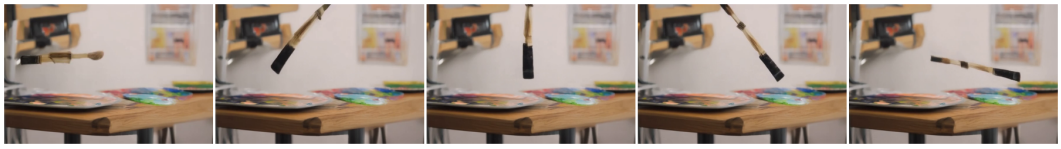


Figure 10: Parabolic Motion with Rotation. The object follows a projectile arc while simultaneously rotating.

similarity, and LPIPS measures perceptual similarity in feature space. For multi-step rollout, we report these metrics at different prediction horizons to analyze how errors accumulate over time.

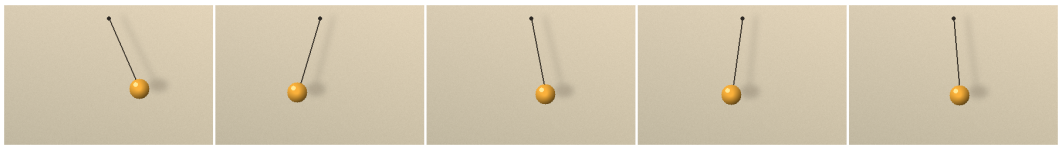


Figure 11: Damped Oscillation. The object undergoes pendulum-like oscillation with decreasing amplitude.

Figure 12: Size Changing. The object changes size over time while remaining visually coherent.

Figure 13: Deformation. The object continuously changes shape while remaining a single coherent object.

Physical consistency. Following NewtonGen, we report the Physical Invariance Score (PIS) to measure whether a physical quantity that should remain invariant over time is stably preserved. Given a physical quantity C, PIS is defined as

$$\mathrm{PIS} = \left(1 + \frac{\sigma_C}{|\mu_C| + \epsilon}\right)^{-1}, \tag{36}$$

where μ_C and σ_C denote the temporal mean and standard deviation of C, and ϵ is a small constant for numerical stability. PIS ranges from 0 to 1, where a larger value indicates better preservation of the corresponding physical invariant.

Different motion types use different physical quantities. For example, uniform motion uses horizontal velocity v_x, uniformly accelerated and decelerated motion use horizontal acceleration a_x, projectile motion uses (v_x, a_y), circular motion and rotation use angular velocity ω, scale variation uses radius change rate Δr, and deformation uses long-axis change rate Δl. When ground-truth physical states are not available, we follow NewtonGen and estimate the relevant quantities from video. We first segment the target object using SAM2 and then estimate velocity, acceleration, angular velocity, and scale variation from object masks, centroids, and shape features. If simulator states are available, we directly compute these quantities in state space to reduce visual estimation noise.

We also report Background Consistency (BC) and Motion Smoothness (MS). BC measures whether the background remains stable during long-horizon prediction, while MS measures the temporal smoothness of predicted motion. These two metrics help distinguish two failure cases: predictions that are physically plausible but visually unstable, and predictions that are visually smooth but physically incorrect.

Method-specific physical diagnostics. Because the central mechanism of LaWM is a DEL-defined variational transition induced by the learned discrete action, we additionally report two diagnostics that directly measure whether the learned dynamics become more physically admissible. The first is the average stationary action residual:

$$R_{\mathrm{stat}} = \frac{1}{H-1} \sum_{k=t+1}^{t+H-1} \|r_k^{\mathrm{SA}}(S)\|_2^2. \tag{37}$$

A smaller R_{stat} means that the predicted latent trajectory is closer to satisfying the local stationary action condition.

The second is relative energy drift:

$$\text{EnergyDrift} = \frac{1}{H} \sum_{k=t+1}^{t+H} \frac{|E_\theta(\hat{q}_k, \hat{v}_k) - E_\theta(\hat{q}_t, \hat{v}_t)|}{|E_\theta(\hat{q}_t, \hat{v}_t)| + \epsilon}, \tag{38}$$

where

$$E_\theta(q, v) = T_\theta(q, v) + V_\theta(q). \tag{39}$$

This metric evaluates whether the learned latent dynamics suffer from long-horizon energy drift.

Embodied prediction metrics. For RoboScape-style embodied scenes, we further report metrics along three dimensions: appearance fidelity, geometric consistency, and action controllability. Appearance fidelity is measured by PSNR and LPIPS. Geometric consistency is measured by AbsRel, δ_1, and δ_2, where AbsRel captures relative depth error and δ_1/δ_2 measure depth prediction accuracy under different thresholds. Action controllability is measured by APSNR, which evaluates whether the predicted future frames respond correctly to changes in the action input. Together with PIS, BC, MS, stationary action residual, and energy drift, these metrics provide a more complete evaluation of visual quality, physical plausibility, geometric consistency, and action-conditioned controllability.

A.5 Ablation Study

A.5.1 Variational Transition Ablation

This section evaluates whether LaWM's improvement comes from using the discrete action principle as the latent transition rule, rather than applying it as a post-hoc trajectory refinement objective. We compare LaWM against a gradient-based trajectory refinement baseline under the same visual encoder, decoder, latent dimensionality, training data, and evaluation protocol. The gradient-based baseline optimizes a completed latent trajectory, whereas LaWM generates each next latent state through the DEL-defined variational transition.

Across most conservative or near-conservative motion categories, LaWM improves over gradient-based refinement in PIS, BC, and MS. This supports the claim that the discrete Euler–Lagrange update provides a more effective structure-preserving rollout mechanism than post-hoc trajectory optimization. In strongly non-conservative or scale-changing cases such as damped oscillation, size changing, and deformation, gradient-based refinement remains stronger on the corresponding PIS metric. We report these cases explicitly because they correspond to dissipative, scale-changing, or deformable dynamics, where forced DEL extensions may be more appropriate than an unforced latent discrete Lagrangian.

The aggregate comparison shows that LaWM improves motion-balanced mPIS from 0.756 to 0.888 and row-wise mPIS from 0.742 to 0.904. LaWM wins on 14 of 17 PIS quantities, all 12 MS quantities, and 9 of 12 BC quantities. The remaining PIS cases where GD refinement is stronger are damped oscillation, size changing, and deformation, which are precisely the motion families where dissipation, scale change, or non-rigid deformation is not fully captured by an unforced discrete Lagrangian. Thus, the ablation supports the role of the DEL-defined transition while also identifying the expected limitation of the current conservative formulation.

The controlled diagnostic separates transition-level structure from post-hoc correction. GD refinement reduces the DEL residual relative to the unconstrained neural transition, but it only slightly improves PIS and leaves a large energy-drift gap. LaWM, by contrast, defines each transition through the DEL condition and therefore achieves lower drift and residual in the controlled latent system. This result should be interpreted as mechanism-level evidence rather than as a replacement for the full visual evaluation in Table 3.

The GD budget sensitivity rules out the explanation that post-hoc refinement simply needs more optimization steps. Increasing the number of GD iterations reduces the optimized residual, but the energy drift remains around 0.265 even at 100 iterations. LaWM reaches substantially lower drift without optimizing a completed trajectory. This supports the conclusion that the advantage comes from using the variational condition as the rollout rule, rather than using it only as a post-hoc correction objective.

Table 3: Ablation comparing gradient-based trajectory refinement with LaWM variational transition. All metrics are higher is better. Values are reported as mean(std) when available. Green indicates the best learned method, and light green indicates the second-best learned method.

Motion Type	Metric	Reference	Sora	Veo3	CogVideoX-5B	Wan2.2	PhyT2V	NewtonGen	Gradient Descent	LaWM
Uniform Motion	PIS-v_x	0.9972	0.6548(0.022)	0.9784(0.006)	0.5392(0.007)	0.6395(0.029)	0.5349(0.014)	0.9830(0.005)	0.9699(0.0195)	**0.9938(0.0045)**
Uniform Motion	BC	1	0.9573(0.003)	0.9491(0.024)	0.9534(0.018)	0.9683(0.027)	0.9612(0.015)	0.9694(0.020)	0.9847(0.0061)	**0.9930(0.0021)**
Uniform Motion	MS	1	0.9926(0.003)	0.9953(0.001)	0.9905(0.005)	0.9939(0.003)	0.9876(0.015)	0.9962(0.003)	0.9958(0.0004)	**0.9993(0.0001)**
Acceleration	PIS-a_x	0.8489	0.3437(0.355)	0.6187(0.308)	0.5458(0.038)	0.3077(0.261)	0.5033(0.011)	0.6568(0.013)	0.4004(0.2294)	**0.8964(0.0275)**
Acceleration	BC	1	0.9495(0.011)	0.9373(0.015)	0.9518(0.037)	0.9695(0.018)	0.9636(0.021)	0.9748(0.012)	0.9838(0.0101)	**0.9960(0.0030)**
Acceleration	MS	1	0.9852(0.011)	0.9909(0.004)	0.9876(0.008)	0.9908(0.005)	0.9822(0.010)	0.9918(0.009)	0.9955(0.0004)	**0.9995(0.0001)**
Deceleration	PIS-a_x	0.8872	0.6162(0.072)	0.6173(0.102)	0.4988(0.014)	0.4705(0.328)	0.5167(0.023)	0.6891(0.007)	0.4477(0.3292)	**0.8701(0.0448)**
Deceleration	BC	1	0.9494(0.026)	0.9295(0.039)	0.9623(0.017)	0.9721(0.012)	0.9622(0.012)	0.9744(0.012)	0.9818(0.0068)	**0.9923(0.0025)**
Deceleration	MS	1	0.9883(0.006)	0.9933(0.003)	0.9787(0.024)	0.9903(0.007)	0.9814(0.014)	0.9947(0.005)	0.9957(0.0004)	**0.9993(0.0001)**
Parabolic Motion	PIS-v_x	0.9988	0.9095(0.014)	0.9042(0.012)	0.7392(0.007)	0.7747(0.126)	0.6370(0.199)	0.9803(0.002)	0.9741(0.0270)	**0.9961(0.0030)**
Parabolic Motion	PIS-a_y	0.9487	0.5723(0.266)	0.7662(0.139)	0.4230(0.028)	0.5571(0.953)	0.3567(0.799)	0.8189(0.014)	0.6983(0.1268)	**0.9443(0.0170)**
Parabolic Motion	BC	1	0.9486(0.023)	0.9514(0.023)	0.9330(0.030)	0.9602(0.028)	0.9436(0.046)	0.9693(0.014)	0.9806(0.0087)	**0.9882(0.0049)**
Parabolic Motion	MS	1	0.9915(0.004)	0.9948(0.002)	0.9856(0.009)	0.9903(0.007)	0.9844(0.011)	0.9967(0.001)	0.9956(0.0005)	**0.9994(0.0000)**
3D Motion	PIS-Δl	0.7388	0.5013(0.005)	0.5932(0.005)	0.3026(0.005)	0.4583(0.005)	0.2911(0.007)	0.6472(0.005)	0.8828(0.2621)	**0.9137(0.2333)**
3D Motion	PIS-v_y	0.9986	0.8481(0.008)	0.8913(0.008)	0.6690(0.003)	0.8384(0.018)	0.6510(0.002)	0.9371(0.007)	0.9359(0.0278)	**0.9850(0.0081)**
3D Motion	BC	1	0.9426(0.017)	0.9410(0.022)	0.9620(0.018)	0.9772(0.008)	0.9629(0.016)	0.9672(0.018)	0.9874(0.0040)	**0.9914(0.0055)**
3D Motion	MS	1	0.9934(0.003)	0.9944(0.003)	0.9945(0.003)	0.9943(0.002)	0.9888(0.012)	0.9954(0.005)	0.9942(0.0013)	**0.9995(0.0001)**
Slope Sliding	PIS-a_x	0.8741	0.4931(0.153)	0.6081(0.157)	0.3533(0.160)	0.3108(0.421)	0.3570(0.354)	0.6312(0.041)	0.3334(0.2556)	**0.8524(0.0383)**
Slope Sliding	PIS-a_y	0.9148	0.4616(0.212)	0.3815(0.092)	0.4731(0.028)	0.3967(0.744)	0.4297(0.569)	0.5840(0.043)	0.2115(0.2838)	**0.8692(0.0560)**
Slope Sliding	BC	1	0.9667(0.013)	0.9631(0.016)	0.9556(0.024)	0.9653(0.017)	0.9568(0.022)	0.9787(0.010)	0.9894(0.0035)	**0.9952(0.0021)**
Slope Sliding	MS	1	0.9919(0.005)	0.9958(0.002)	0.9903(0.006)	0.9912(0.005)	0.9829(0.014)	0.9971(0.001)	0.9958(0.0004)	**0.9994(0.0001)**
Circular Motion	PIS-ω	0.9933	0.8393(0.010)	0.8932(0.007)	0.7726(0.026)	0.4677(0.006)	0.6391(0.322)	**0.9788(0.018)**	0.9481(0.0217)	0.9562(0.0346)
Circular Motion	BC	1	0.9684(0.012)	0.9711(0.010)	0.9842(0.013)	0.9745(0.016)	0.9677(0.027)	0.9812(0.007)	0.9841(0.0032)	**0.9906(0.0026)**
Circular Motion	MS	1	0.9949(0.001)	0.9960(0.001)	0.9979(0.001)	0.9949(0.004)	0.9974(0.002)	0.9980(0.001)	0.9960(0.0003)	**0.9995(0.0001)**
Rotation	PIS-ω	0.9836	0.4267(0.099)	0.5285(0.436)	0.6596(0.023)	0.3425(0.172)	0.7842(0.304)	0.8838(0.038)	0.7209(0.1133)	**0.9436(0.0197)**
Rotation	BC	1	0.9543(0.030)	0.9650(0.018)	0.9397(0.025)	0.9620(0.010)	0.9375(0.028)	0.9700(0.008)	0.9818(0.0051)	**0.9850(0.0052)**
Rotation	MS	1	0.9900(0.006)	0.9942(0.003)	0.9795(0.028)	0.9909(0.007)	0.9878(0.006)	0.9958(0.002)	0.9953(0.0005)	**0.9996(0.0000)**
Para. w/ Rotation	PIS-v_x	0.9990	0.5797(0.150)	0.7029(0.197)	0.6488(0.031)	0.6558(0.175)	0.7689(0.039)	0.9446(0.008)	0.9691(0.0252)	**0.9943(0.0040)**
Para. w/ Rotation	PIS-a_y	0.9657	0.4903(2.581)	0.5603(1.012)	0.2614(0.127)	0.4331(1.982)	0.2879(0.046)	0.5614(0.028)	0.6533(0.0858)	**0.9267(0.0290)**
Para. w/ Rotation	PIS-ω	0.9829	0.6522(0.556)	0.9019(0.119)	0.3380(0.199)	0.3474(0.334)	0.4119(0.105)	0.9289(0.029)	0.6579(0.1997)	**0.9645(0.0147)**
Para. w/ Rotation	BC	1	0.9532(0.016)	0.9583(0.018)	0.9567(0.018)	0.9617(0.027)	0.9675(0.020)	0.9786(0.009)	0.9713(0.0066)	**0.9869(0.0057)**
Para. w/ Rotation	MS	1	0.9889(0.005)	0.9952(0.001)	0.9841(0.018)	0.9908(0.005)	0.9921(0.003)	0.9969(0.001)	0.9951(0.0006)	**0.9995(0.0000)**
Damped Oscillation	PIS-a_y	0.9402	0.4418(0.364)	0.3516(0.482)	0.3083(0.055)	0.3494(0.395)	0.2841(0.042)	0.5240(0.017)	**0.9780(0.1077)**	0.8153(0.1076)
Damped Oscillation	BC	1	0.9738(0.013)	0.9666(0.011)	0.9699(0.007)	0.9715(0.014)	0.9708(0.010)	0.9743(0.012)	**0.9975(0.0019)**	0.9935(0.0026)
Damped Oscillation	MS	1	0.9909(0.014)	0.9958(0.001)	0.9853(0.005)	0.9919(0.009)	0.9867(0.003)	0.9968(0.001)	0.9959(0.0004)	**0.9990(0.0000)**
Size Changing	PIS-Δr	0.8501	0.2840(0.010)	0.4167(0.006)	0.5774(0.007)	0.1972(0.022)	0.4010(0.011)	0.6362(0.010)	**0.9712(0.1410)**	0.8263(0.3048)
Size Changing	BC	1	0.9507(0.025)	0.9548(0.017)	0.9636(0.019)	0.9735(0.018)	0.9666(0.022)	0.9669(0.015)	**0.9970(0.0031)**	0.9928(0.0037)
Size Changing	MS	1	0.9916(0.003)	0.9955(0.002)	0.9926(0.007)	0.9889(0.008)	0.9925(0.004)	0.9955(0.002)	0.9961(0.0002)	**0.9994(0.0001)**
Deformation	PIS-Δl	0.9247	0.3626(0.004)	0.3466(0.017)	0.3550(0.002)	0.3515(0.043)	0.3601(0.003)	0.5492(0.005)	**0.8561(0.2866)**	0.6141(0.3621)
Deformation	BC	1	0.9553(0.039)	0.9058(0.052)	0.9462(0.018)	0.9347(0.042)	0.9211(0.010)	0.9475(0.025)	**0.9974(0.0031)**	0.9911(0.0034)
Deformation	MS	1	0.9941(0.004)	0.9940(0.006)	0.9935(0.009)	0.9903(0.009)	0.9867(0.001)	0.9957(0.001)	0.9954(0.0005)	**0.9994(0.0000)**

Table 4: Aggregate ablation summary comparing post-hoc GD refinement with LaWM. Motion-balanced mPIS first averages PIS quantities within each motion type and then averages across motion types. The aggregate is computed from Table 3.

Method	Motion-balanced mPIS ↑	Row-wise mPIS ↑	mBC ↑	mMS ↑	PIS wins
GD Refinement	0.756	0.742	0.986	0.996	3 / 17
LaWM	**0.888**	**0.904**	**0.991**	**0.999**	**14 / 17**

Table 5: Controlled latent mechanism diagnostic isolating the role of the transition rule. This diagnostic is not a full visual rollout evaluation; it is designed to test whether the DEL-defined transition provides the expected structure-preserving bias in a controlled latent system. Higher is better for PIS; lower is better for energy drift and DEL residual.

Method	Rollout Rule	PIS@64 ↑	Energy Drift ↓	DEL Residual ↓
Neural transition	unconstrained predictor	0.906	0.272	1.06e-04
GD refinement	post-hoc trajectory optimization	0.907	0.266	1.68e-05
LaWM	DEL variational step	**0.997**	**0.028**	**1.31e-08**

A.5.2 Latent Lagrangian Architecture Ablation

We next evaluate the architectural choices introduced in LaWM's latent variational dynamics. These ablations isolate the contribution of the Lagrangian transition, the $T - V$ decomposition, the positive diagonal mass model, the sequence-level context variable, the DEL consistency loss, the mass-

Table 6: GD iteration-budget sensitivity in the controlled latent diagnostic. Increasing the post-hoc optimization budget reduces the DEL residual but does not close the energy-drift gap to the direct DEL update. Runtime is reported only as a controlled micro-benchmark.

Method	GD Iterations	Energy Drift ↓	DEL Residual ↓	Runtime (ms) ↓
GD refinement	5	0.268	1.45e-05	0.0009
GD refinement	20	0.266	1.77e-05	0.0048
GD refinement	50	0.265	1.68e-05	0.0096
GD refinement	100	0.263	1.66e-05	0.0231
LaWM	direct	**0.028**	**1.31e-08**	0.0005

conditioning regularizer, and the finite unrolled DEL solver. To avoid confounding architectural effects with visual rendering artifacts, we evaluate these variants in a controlled latent mechanical diagnostic with varying mass, stiffness, and sequence context. This diagnostic is intended to test the internal structure of the latent dynamics, rather than to replace full visual evaluation.

Table 7: Latent Lagrangian architecture ablation in a controlled latent mechanical diagnostic. All variants are evaluated at horizon $H = 128$ over eight seeds. Higher is better for PIS; lower is better for energy drift, DEL residual, and rollout MSE.

Variant	PIS@128 ↑	Energy Drift ↓	DEL Residual ↓	Rollout MSE ↓
LaWM	**0.980**	**0.020**	**4.36e-12**	**1.54e-06**
Neural Dynamics	0.485	0.723	2.58e-06	7.15e-01
Direct Scalar Lagrangian	0.970	0.030	3.12e-08	4.97e-04
Identity Mass	0.972	0.029	8.77e-08	5.26e-02
Fixed Diagonal Mass	0.964	0.037	7.78e-08	5.47e-02
No Context	0.967	0.034	1.11e-07	4.95e-02
No DEL Loss	0.766	0.266	1.24e-06	1.01e+00
No Mass Regularization	0.961	0.040	1.24e-07	1.15e-03

The complete LaWM model achieves the strongest long-horizon physical consistency in the controlled latent diagnostic, with PIS@128 of 0.980 and energy drift of 0.020. Replacing the variational transition with a neural dynamics model causes the largest degradation, reducing PIS@128 to 0.485 and increasing energy drift to 0.723. This confirms that the learned discrete action is not merely an auxiliary regularizer, but the mechanism that defines the rollout.

The structured Lagrangian parameterization also contributes to stability. The direct scalar Lagrangian variant remains competitive in PIS, but increases energy drift and rollout error relative to the full model. Replacing the learnable mass with identity or fixed diagonal masses preserves the coarse variational form, but substantially increases long-horizon rollout MSE. This indicates that state- and context-dependent inertia is important for heterogeneous latent dynamics. Removing the context variable similarly degrades PIS, energy drift, and rollout MSE. Removing the DEL loss produces a much larger degradation, suggesting that the training objective must explicitly encourage variational consistency for the finite unrolled solver to remain stable. Removing the mass-conditioning regularizer has a smaller but consistent effect, increasing both energy drift and rollout error.

Table 8: DEL solver iteration ablation for the unrolled differentiable residual solver. All variants use the same latent Lagrangian architecture and differ only in the number of solver iterations N.

Variant	PIS@128 ↑	Energy Drift ↓	DEL Residual ↓	Rollout MSE ↓
DEL Solver, $N = 1$	0.820	0.199	8.13e-07	6.62e-01
DEL Solver, $N = 2$	0.951	0.051	1.56e-07	9.95e-02
DEL Solver, $N = 4$	0.979	0.027	4.59e-09	2.75e-03
DEL Solver, $N = 8$	**0.980**	**0.020**	**4.36e-12**	**1.54e-06**

The solver-iteration ablation evaluates the finite unrolled DEL solve used by LaWM. One or two correction steps leave substantial residual and long-horizon rollout error. Four steps are sufficient to recover stable PIS and energy drift, while eight steps further reduces the DEL residual and rollout MSE toward numerical precision. This supports using a small fixed number of differentiable solver

iterations: the first few iterations provide the main stability gain, and additional iterations mainly improve numerical consistency.

A.5.3 Long-Horizon Stability Analysis

We evaluate whether the advantage of the variational transition persists as the rollout horizon increases. To avoid relying only on an idealized latent diagnostic, we organize the evidence into three levels. First, we run a state-space long-horizon audit on all 12 benchmark motion families. Second, we render a visual subset and evaluate PIS on the generated mask videos. Third, we include a controlled latent diagnostic only as a mechanism-level stress test of the DEL-induced transition. This separation is important: the controlled diagnostic isolates numerical structure preservation, while the state-space and visual-subset audits are closer to the benchmark evaluation setting.

Table 9: State-space long-horizon audit on the 12 benchmark motion families. This audit uses the benchmark state variables directly and does not render videos. Higher is better for PIS; lower is better for normalized state RMSE.

Method	mPIS@64 ↑	mPIS@128 ↑	mPIS@200 ↑	State RMSE@64 ↓	State RMSE@128 ↓	State RMSE@200 ↓
Neural state transition	0.6093	0.5692	0.5413	1.431	5.237	12.373
GD-refined state transition	0.6169	0.5760	0.5485	1.433	5.234	12.375
LaWM state rollout	**0.8264**	**0.8442**	**0.8560**	**0.0103**	**0.0105**	**0.0101**

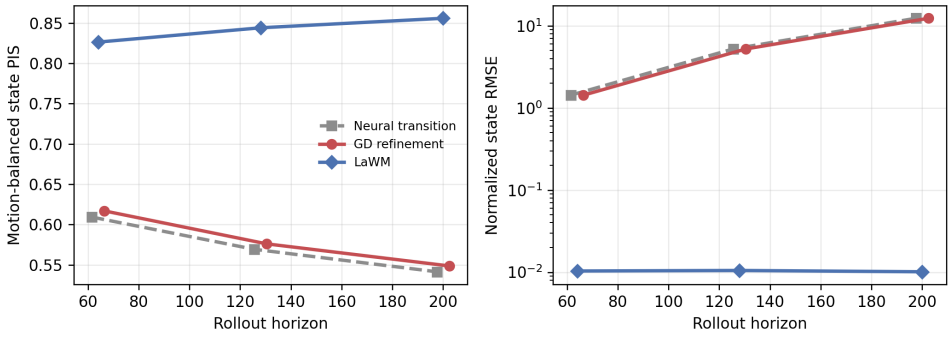


Figure 14: State-space long-horizon audit on the 12 benchmark motion families. Left: motion-balanced state PIS. Right: normalized state RMSE. LaWM remains stable as the rollout horizon increases, while both the unconstrained state transition and post-hoc GD refinement degrade.

The state-space audit provides the main long-horizon evidence because it uses the same benchmark motion families as the visual evaluation. The unconstrained state transition degrades from mPIS 0.6093 at $H = 64$ to 0.5413 at $H = 200$. Post-hoc GD refinement provides a small improvement, increasing mPIS from 0.6093 to 0.6169 at $H = 64$ and from 0.5413 to 0.5485 at $H = 200$, but it does not prevent long-horizon degradation. In contrast, LaWM remains stable across horizons, with mPIS increasing from 0.8264 to 0.8560 and normalized state RMSE remaining near 0.010. This supports the claim that applying the variational update as the rollout rule is more reliable than correcting a completed trajectory after rollout.

Table 10: Full visual subset sanity check at horizon $H = 128$. We render three representative benchmark motion families and evaluate PIS on the generated mask videos. Higher is better.

Motion	Metric	Neural State	GD-refined State	LaWM State
Parabolic Motion	PIS-v_x	0.979	**0.980**	0.978
Parabolic Motion	PIS-a_y	0.283	0.303	**0.819**
Circular Motion	PIS-ω	0.846	0.845	**0.980**
Deformation	PIS-Δl	0.443	0.430	**0.925**
Mean over PIS metrics		0.638	0.639	**0.925**

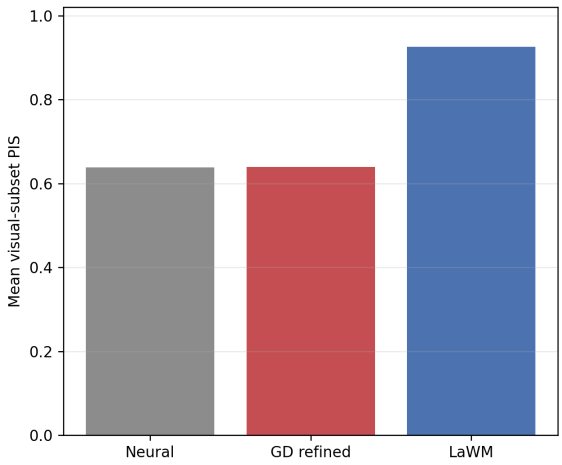


Figure 15: Full visual subset sanity check at $H = 128$. We render parabolic motion, circular motion, and deformation from the audited state rollouts and evaluate PIS on mask videos. LaWM improves mean PIS over both the neural state transition and the GD-refined state transition.

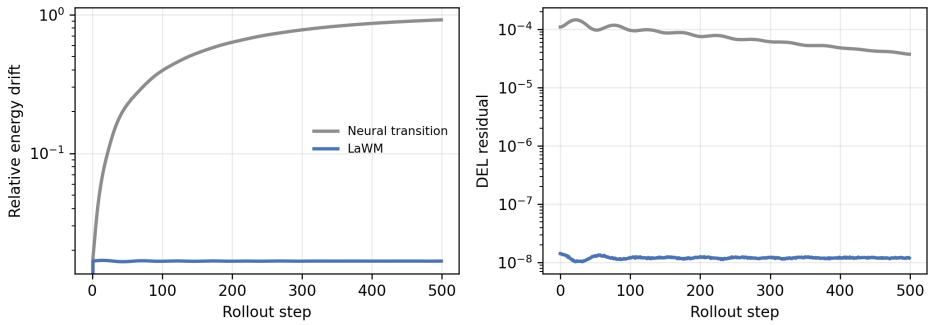
The visual subset confirms that the long-horizon advantage is not only an artifact of evaluating raw state variables. After rendering representative benchmark motions and evaluating PIS on the generated mask videos, LaWM obtains mean PIS of 0.925, compared with 0.638 for the neural state transition and 0.639 for the GD-refined state transition. GD refinement slightly improves the mean over the neural state transition, but the improvement is marginal, consistent with the state-space audit. This result is intentionally more conservative than the idealized controlled latent diagnostic, but it is more directly tied to the visual evaluation pipeline.

Table 11: Controlled latent mechanism diagnostic. This diagnostic isolates the numerical behavior of the transition rule in a conservative latent system and is not a replacement for full visual rollout evaluation. Higher is better for PIS; lower is better for drift and DEL residual.

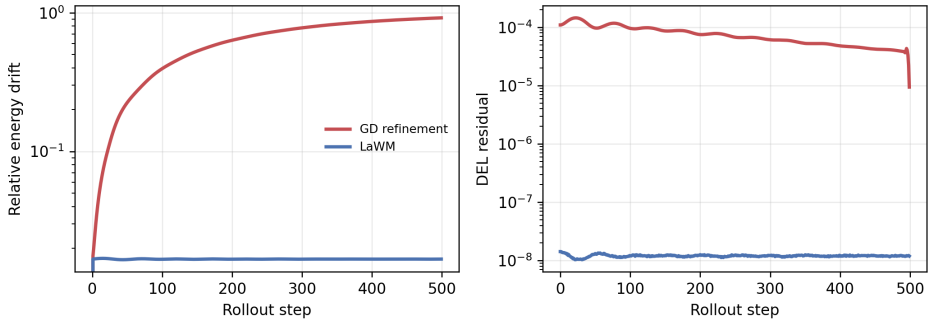
Method	**PIS@200** ↑	**PIS@500** ↑	**Drift@200** ↓	**Drift@500** ↓	**DEL@500** ↓
Neural transition	0.753	0.543	0.662	0.937	3.74e-05
GD refinement	0.774	0.590	0.633	0.918	9.43e-06
LaWM	**0.998**	**0.999**	**0.0167**	**0.0167**	**1.19e-08**

The controlled latent diagnostic explains why the state-space and visual-subset audits favor the variational transition. In this diagnostic, the system is deliberately conservative and therefore matches the assumptions of the DEL-induced rollout. Under this setting, LaWM remains close to the controlled-system invariance ceiling, while the neural transition and GD refinement accumulate substantial energy drift over long horizons. GD refinement reduces the final DEL residual compared with the unconstrained transition, but it does not close the energy-drift gap. The result should not be read as a claim that full visual rollouts remain perfect for 500 frames. Instead, it shows that the proposed transition mechanism has the expected structure-preserving bias when evaluated in isolation.

Overall, these three levels of evidence support the same conclusion. The benchmark state-space audit shows stable long-horizon behavior across all 12 motion families, the rendered visual subset confirms that the trend is visible after video generation and PIS evaluation, and the controlled latent diagnostic provides a mechanistic explanation for why the DEL-defined transition is more stable than post-hoc trajectory refinement.



(a) Neural transition vs. LaWM



(b) GD refinement vs. LaWM

Figure 16: Controlled latent mechanism diagnostic extended to 500 rollout steps. LaWM keeps relative energy drift nearly constant and DEL residual near numerical precision, while the unconstrained neural transition and post-hoc GD refinement accumulate large energy drift over long horizons.