ONEWORLD: LEARNING CONSISTENT PHYSICS ACROSS ACTIONS IN WORLD MODELS

Anonymous authors
Paper under double-blind review

ABSTRACT

Action-conditioned video world models aim to predict scene evolution under different actions, a capability that is essential for reliable planning, decision-making, and interaction in dynamic environments. However, futures generated independently from the same initial scene may each appear plausible while implying incompatible physical properties, such as friction or mass. This inconsistency can lead to contradictory predictions across interventions, making it difficult for the model to maintain a coherent understanding of the underlying world and limiting its reliability for planning and decision-making. To address these issues, we propose **OneWorld**, a shared-mechanism counterfactual generation framework that jointly models multiple action-conditioned futures under a common latent physical mechanism. A physical mechanism interpreter first infers a distribution over latent mechanisms from each action–outcome branch. These distributions are then aggregated into shared-world evidence, which captures whether the branches admit a common physical explanation while accounting for uncertainty in less informative branches. This evidence constrains flow training and guides sampling, encouraging consistency in the underlying physical mechanism while preserving the distinct outcomes induced by different actions. We further introduce a multi-intervention evaluation protocol in controlled environments, following the interaction settings of ACWM-Phys, to assess whether generated futures can be jointly explained by the same physical parameters, alongside standard measures of single-rollout prediction quality. Experiments in these environments show that OneWorld improves cross-intervention physical consistency while maintaining competitive single-rollout prediction quality. *The source code will be released.*

1 INTRODUCTION

Action-conditioned video world models aim to predict how a scene evolves under different actions, a capability that is increasingly important for planning, decision-making, and interaction in dynamic environments (Hafner et al., 2019; 2020; 2021; Hansen et al., 2022; 2024; Hafner et al., 2025; Yang et al., 2024). Given an initial visual observation and a candidate action sequence, these models predict the resulting future video, allowing an agent to compare possible outcomes before choosing how to act (Kaiser et al., 2020; Schrittwieser et al., 2020; Bruce et al., 2024; Wu et al., 2024; Huang et al., 2026; Xue et al., 2026). Such comparisons implicitly assume that alternative futures correspond to different interventions on the same underlying world. The actions may change, and their consequences should differ, but action-independent physical properties such as object mass, surface friction, or material stiffness should remain consistent across the predicted futures.

Current action-conditioned world models typically generate each future independently under its corresponding action sequence and do not explicitly enforce consistency across alternative futures from the same initial world (Bruce et al., 2024; Gupta et al., 2023; Wu et al., 2024; Guo et al., 2026; Huang et al., 2026; Shi et al., 2026; Xue et al., 2026). Consequently, predictions can each appear plausible in isolation while implying mutually incompatible physical properties. For example, two cube trajectories under different pushes may each be individually reasonable, yet one may require a high-friction surface while the other is only compatible with a low-friction one (Wu et al., 2015; Agrawal et al., 2016; Li et al., 2020). Figure 1 illustrates this failure mode with ACWM-DiT under multiple actions from fixed initial observations. We refer to this as *shared-world inconsistency* and seek *shared-world consistency*, where action-dependent futures preserve distinct trajectories while

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



Figure 1: **Motivating probes of action-conditioned prediction.** ACWM-DiT predictions under different actions from the same initial observation within each scene, illustrating the need to evaluate whether alternative futures remain compatible with a shared physical world.

admitting a common physical explanation. We quantify this property by comparing branch-wise physical fits with a joint fit constrained to share one physical configuration, with the additional fitting error defining the *shared-world gap*. The nonzero gaps for ACWM-DiT in Table 1 show that individually plausible rollouts can remain inconsistent as outcomes of one shared physical world.

Modeling shared-world consistency introduces two key challenges. First, the physical properties implied by a prediction cannot be inferred from the outcome alone; they must be interpreted jointly with the action that produced it (Agrawal et al., 2016; Li et al., 2020; Baradel et al., 2020). The displacement of a pushed object, for instance, depends on both the applied action and the physical properties of the object and surface. Second, different interventions provide unequal amounts of information about those properties. A branch in which an object is barely disturbed may remain compatible with many physical configurations, whereas a strong interaction can sharply constrain the plausible mechanisms (Wu et al., 2015; Li et al., 2020). Consequently, physically consistent branches need not induce identical beliefs about the world: they may support the same mechanism with different levels of uncertainty.

To address these challenges, we propose **OneWorld**, a shared-mechanism counterfactual generation framework that jointly models multiple action-conditioned futures through a common latent physical mechanism. A *physical mechanism interpreter* first maps each action–outcome branch to a distribution over latent mechanisms. We then introduce a *shared-world evidence score* that aggregates these branch distributions relative to an initial-scene prior. The score measures whether the branches retain common support for a shared mechanism while allowing less informative interventions to remain uncertain. We first train the interpreter using same-world and mismatched-world groups, and then freeze it to provide a physical compatibility signal for both flow training and sampling-time guidance. In this way, OneWorld couples the physical interpretations of different branches without forcing their predicted videos or mechanism posteriors to become identical.

We evaluate OneWorld on controlled Push Cube, Push Rope, and Pour Water environments following the corresponding interaction settings of ACWM-Phys (Xue et al., 2026). These environments span rigid-body, deformable-object, and particle-based dynamics and allow multiple interventions to be executed from the same underlying physical configuration. We evaluate both conventional single-rollout prediction quality and whether the resulting futures can be jointly explained by one

physical configuration. Relative to ACWM-DiT, OneWorld reduces the mean shared-world gap by **84.3%** while improving mean M-MSE by approximately 7.7×10^{-4}.

We make the following contributions.

- We formulate *shared-world consistency*: action-dependent futures generated from one initial world should admit a common physical explanation.
- We introduce a physical mechanism interpreter and a prior-corrected shared-world evidence score that assess compatibility across interventions while accounting for unequal uncertainty.
- We use the shared-world evidence signal to couple action-conditioned predictions through shared-world flow training and sampling-time flow guidance.
- We introduce an independent simulator-based evaluation protocol that separates individual physical fit from cross-intervention shared-world consistency, and demonstrate substantial consistency gains across three controlled physical environments.

2 Related Work

Action-conditioned video world models. Diffusion and flow-matching models provide the generative foundation for high-capacity visual prediction (Ho et al., 2020; Song et al., 2021a;b; Rombach et al., 2022; Ho et al., 2022; Voleti et al., 2022; Lipman et al., 2023; Peebles & Xie, 2023; Yu et al., 2023). World models extend such predictors to planning and interaction by conditioning future observations on actions (Hafner et al., 2019; 2020; 2021; Kaiser et al., 2020; Schrittwieser et al., 2020; Hansen et al., 2022; 2024; Bruce et al., 2024; Yang et al., 2024; Wu et al., 2024; Huang et al., 2026; Guo et al., 2026). Recent systems increasingly model long-horizon, visually rich dynamics, but their training objectives still primarily supervise the quality of each action-conditioned rollout individually (Denton & Fergus, 2018; Wu et al., 2021; Babaeizadeh et al., 2021; Gupta et al., 2023; Guo et al., 2025; Wang et al., 2025b).

Physical reasoning and evaluation. Physical priors and post-training objectives have been introduced to improve physical behavior in generated videos (Wu et al., 2015; Agrawal et al., 2016; Battaglia et al., 2016; Li et al., 2020; Sanchez-Gonzalez et al., 2020; Pfaff et al., 2021; Wang et al., 2025a; Yuan et al., 2026; Wang et al., 2026), while benchmarks evaluate whether generated motion follows plausible physical dynamics (Bakhtin et al., 2019; Yi et al., 2020; Huang et al., 2024; Bansal et al., 2025; Li et al., 2025; Bansal et al., 2026; Motamed et al., 2026; Xue et al., 2026). ACWM-Phys provides the interaction settings and action-conditioned backbone used in our controlled experiments. Existing evaluation is largely single-rollout, whereas our protocol asks whether several futures from the same initial world can be explained by one shared physical configuration.

Counterfactual and multi-branch consistency. CoPhy reasons about latent physical factors for counterfactual prediction (Baradel et al., 2020; Yang et al., 2021; Ahmed et al., 2021), and Multiverse Mechanica studies causal consistency across parallel worlds with matched initial conditions (Ness et al., 2026). CoCo improves action responsiveness through counterfactual constraints (Shi et al., 2026), while Twin Rollouts couples factual and counterfactual branches through shared prefixes and future exogenous noise (Ma et al., 2026). OneWorld instead couples multiple action-conditioned futures through the compatibility of their inferred physical explanations, without requiring identical branch posteriors or shared sampling noise.

3 Method

As illustrated in Figure 2, OneWorld generates multiple action-conditioned futures from the same initial scene and couples them through a shared physical interpretation. We first formulate counterfactual generation from a common initial world and introduce the underlying action-conditioned flow model (Section 3.1). We then learn a physical mechanism interpreter that maps each action–outcome branch to a latent mechanism distribution and uses their shared support to measure cross-branch compatibility (Section 3.2). Finally, we use this shared-world evidence to optimize the video model during both training and sampling (Section 3.3).

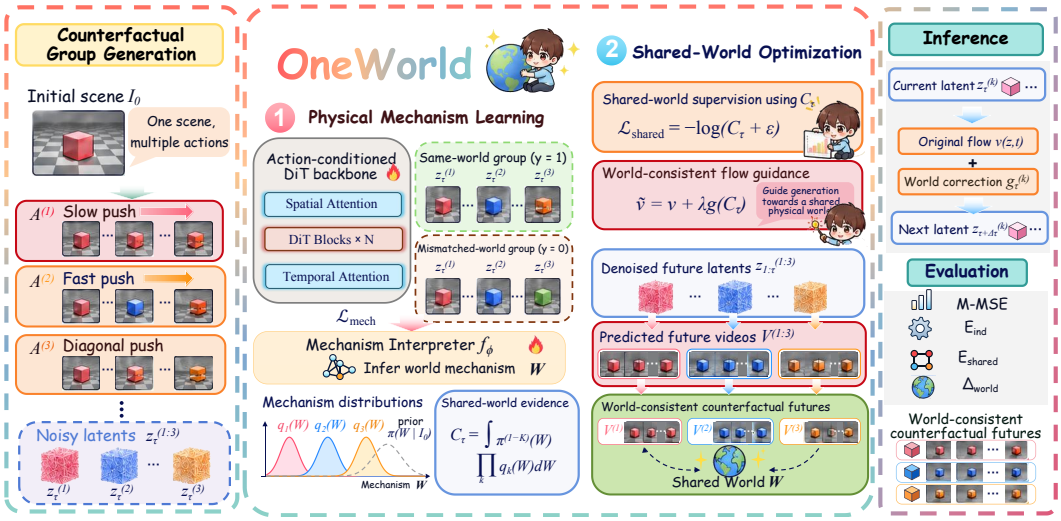


Figure 2: Overview of **OneWorld**. Given the same initial scene and multiple action sequences, OneWorld predicts a future for each intervention branch. A physical mechanism interpreter infers a mechanism distribution from each action–outcome pair. The shared-world evidence score assesses their compatibility and provides the signal for shared-world flow training and world-consistent flow guidance.

3.1 Task Formulation and Counterfactual Generation

We begin by formalizing the multi-intervention prediction setting considered throughout OneWorld. The key requirement is that different actions may produce different future trajectories, while all branches generated from the same initial world remain compatible with one shared physical mechanism. We then instantiate each intervention branch with an action-conditioned latent flow model that provides the future representation used by the subsequent physical mechanism learning stage.

Counterfactual formulation. Let I_0 denote the initial observation of a scene. We consider K action interventions from the same initial world, each defining an intervention branch. For branch k, $A^{(k)} = \{a_t^{(k)}\}_{t=0}^{T-1}$ is the action sequence and $V^{(k)} = \{I_t^{(k)}\}_{t=1}^{T}$ is the corresponding future video. We refer to these K branches collectively as a *counterfactual group*. We use W to denote a latent mechanism variable representing action-independent properties of the world, such as those associated with friction, mass, or stiffness. Its coordinates need not correspond to named physical quantities. We distinguish W from the explicit simulator parameters w used for evaluation in Section 4.1. Different interventions are expected to produce different outcomes, but all branches originating from the same initial world should be compatible with the same realization of W. This requirement motivates the following shared-mechanism factorization

$$p\left(V^{(1:K)} \mid I_0, A^{(1:K)}\right) = \int p(W \mid I_0) \prod_{k=1}^{K} p\left(V^{(k)} \mid I_0, A^{(k)}, W\right) dW, \tag{1}$$

where $A^{(1:K)}$ and $V^{(1:K)}$ denote all intervention sequences and their future videos. Equation 1 assumes conditional independence of the futures given I_0, their actions, and a shared W. For the pushing example, the trajectories under different actions should be compatible with a common physical configuration, including properties such as friction and mass.

Action-conditioned flow model. We build OneWorld on an action-conditioned latent flow model following ACWM-DiT (Xue et al., 2026). Rather than directly parameterizing the conditional factors in Eq. 1, we use an auxiliary mechanism interpreter to encourage shared-world consistency during flow training and sampling. Let $z_\tau^{(k)}$ denote the future latent of branch k at flow time $\tau \in [0, 1]$, where $\tau = 0$ corresponds to noise and $\tau = 1$ to the clean future. For a linear flow path,

$z_\tau^{(k)} = (1-\tau)\epsilon^{(k)} + \tau z_1^{(k)}$, where $\epsilon^{(k)}$ is Gaussian noise and $z_1^{(k)}$ is the clean future latent (Lipman et al., 2023). The pretrained action-conditioned model predicts a velocity $v_\theta(z_\tau^{(k)}, \tau, I_0, A^{(k)})$ for each branch. OneWorld keeps this branch-specific action prediction while introducing cross-branch constraints on the physical mechanism.

At flow time τ, the current vector field provides an estimate of the clean future latent,

$$\hat{z}_{1,\tau}^{(k)} = z_\tau^{(k)} + (1-\tau) v_\theta \left(z_\tau^{(k)}, \tau, I_0, A^{(k)} \right). \tag{2}$$

Under the linear flow path, Eq. 2 recovers the clean future when the predicted velocity is exact. We use this estimate rather than the noisy latent itself because physical properties are expressed through the predicted outcome of the intervention.

3.2 Physical Mechanism Learning

The first stage of OneWorld learns a physical compatibility model across intervention branches. Given the action and predicted outcome of each branch, we infer a distribution over latent mechanisms and aggregate these distributions into a shared-world evidence score. The interpreter is learned from same-world and mismatched-world counterfactual groups, allowing physical compatibility to emerge from relational supervision without requiring explicit physical-parameter labels.

Mechanism inference and shared-world evidence. We infer the physical properties implied by a prediction jointly with the action that produced it. Using the clean-future estimate in Eq. 2, a physical mechanism interpreter q_ϕ maps each branch to a distribution over the latent mechanism W,

$$q_\phi^{(k)}(W) = q_\phi \left(W \mid I_0, A^{(k)}, \hat{z}_{1,\tau}^{(k)}, \tau \right). \tag{3}$$

Conditioning on the action disambiguates similar motions produced by different interventions, while τ allows uncertainty to reflect the reliability of the current future estimate.

Let $\pi_\phi(W \mid I_0)$ denote the mechanism prior before observing any intervention outcome. We define the shared-world evidence of a counterfactual group as

$$\mathcal{C}_\tau = \int \pi_\phi(W \mid I_0)^{1-K} \prod_{k=1}^{K} q_\phi^{(k)}(W)\, dW. \tag{4}$$

A high $\mathcal{C}_\tau$ indicates common posterior support relative to the initial-scene prior, while the factor π_ϕ^{1-K} removes repeated prior contributions from the branch posteriors. Under exact Bayesian posteriors and conditional independence, Eq. 4 becomes an evidence ratio between a shared-mechanism model and independently drawn mechanisms (Appendix A); here it is used as a learned compatibility score. This formulation also accommodates interventions with unequal amounts of physical information. If an uninformative branch satisfies $q_\phi^{(j)}(W) = \pi_\phi(W \mid I_0)$, its contribution cancels one prior factor and leaves the score for the remaining $K-1$ branches unchanged (Appendix A.2). We implement this behavior by parameterizing each branch posterior as an additive evidence update to the initial-scene prior in Gaussian natural-parameter space, allowing weakly informative interventions to remain close to the prior while informative ones contribute sharper mechanism estimates. The parameterization and numerical estimation are detailed in Appendix D.

Relational mechanism learning. We learn the mechanism interpreter by contrasting groups that share one physical world with groups whose branches originate from different physical configurations. A same-world group contains different action branches generated from the same initial state and hidden physical configuration, whereas a mismatched-world group keeps the visible initial scene matched but combines branches from different physical configurations. This supervision encourages the interpreter to separate variation induced by actions from variation in the underlying world.

Let $\mathcal{B}$ denote a counterfactual group, with $y = 1$ for same-world groups and $y = 0$ for mismatched-world groups. Using $s(\mathcal{B}) = \log \mathcal{C}_\tau$ as the compatibility score, we optimize

$$\mathcal{L}_{\text{mech}} = -\mathbb{E}_{\mathcal{B}} \left[y \log \sigma(s(\mathcal{B})) + (1-y) \log \left(1 - \sigma(s(\mathcal{B}))\right) \right], \tag{5}$$

where σ is the sigmoid. The supervision acts only at the group level, so the latent coordinates are not required to correspond to named physical quantities.

The interpreter and initial-scene prior are trained before the video model. During this stage, ACWM-DiT is frozen and τ is sampled across the flow trajectory, exposing the interpreter to clean-future estimates with different levels of uncertainty. After training, the interpreter and prior are fixed and used as the compatibility criterion for shared-world optimization. Further details are given in Appendix D.5.

3.3 Shared-World Optimization

The second stage uses the learned physical compatibility model to improve the action-conditioned generator. We keep the mechanism interpreter fixed and reuse the same shared-world evidence at two points in generation. During training, it provides a group-level objective that favors futures admitting a common physical explanation. During sampling, its gradient supplies a trajectory-level correction that preserves this compatibility as the future latents are progressively denoised.

Shared-world flow training. After mechanism learning, we freeze both the interpreter and the initial-scene prior so that the compatibility criterion remains fixed while the video model is updated. For each same-world group, the branch-wise clean-future estimates are interpreted jointly and optimized to yield high shared-world evidence,

$$\mathcal{L}_{\text{shared}} = -\mathbb{E}\left[\log\left(\mathcal{C}_\tau + \epsilon\right)\right], \tag{6}$$

where ϵ ensures numerical stability. This loss couples the branches through their physical interpretations rather than by directly matching their videos, latents, or posterior parameters.

The complete video-model objective is

$$\mathcal{L} = \mathcal{L}_{\text{flow}} + \lambda_{\text{shared}}\mathcal{L}_{\text{shared}}, \tag{7}$$

where branch-wise flow matching $\mathcal{L}_{\text{flow}}$ preserves supervision for the action-specific future, while $\mathcal{L}_{\text{shared}}$ discourages sets of predictions that cannot be explained by one common mechanism. Freezing the interpreter prevents the compatibility model from adapting to the generator during this stage, but does not block optimization of the video model. Gradients from the shared-world loss pass through the interpreter inputs and the clean-future estimate in Eq. 2 to the action-conditioned vector field.

World-consistent flow guidance. The same shared-world evidence is further used to steer the sampling trajectory. For branch k, we compute

$$g_\tau^{(k)} = \nabla_{z_\tau^{(k)}} \log\left(\mathcal{C}_\tau + \epsilon\right), \tag{8}$$

which gives the local latent-space direction that most increases the compatibility of the current counterfactual group. Because $\mathcal{C}_\tau$ depends jointly on all branch posteriors, the correction for one branch is informed by the physical explanation supported by the others.

We modify the original action-conditioned velocity as

$$\widetilde{v}_\theta^{(k)} = v_\theta\left(z_\tau^{(k)}, \tau, I_0, A^{(k)}\right) + \lambda_\tau g_\tau^{(k)}, \tag{9}$$

where λ_τ controls the strength of the world-consistent correction. The original velocity remains responsible for action-conditioned prediction, while the guidance term provides a comparatively small group-level adjustment. We apply weak guidance early in the trajectory, when clean-future estimates are uncertain, and increase its influence as generation becomes more informative. The schedule and gradient approximation are given in Appendix D.4. At inference time, branches share only the initial observation and compatibility signal; their actions and Gaussian noise remain independent, so the method preserves distinct action-conditioned futures while encouraging a shared physical explanation.

4 Experiments

We evaluate shared-world consistency together with single-rollout prediction quality, compare against alternative coupling strategies, and study the contributions and sensitivity of the proposed components.

4.1 Experimental Setup

Benchmarks and baselines. Following the ACWM-Phys interaction settings (Xue et al., 2026), we evaluate on Push Cube, Push Rope, and Pour Water, covering rigid-body, deformable, and fluid dynamics. Same-world groups vary actions while sharing the same initial state and hidden physical configuration, whereas mismatched-world groups keep the visible initial scene fixed but vary the hidden configuration. We compare against ACWM-DiT (Xue et al., 2026), Vid2World (Huang et al., 2026), CoCo (Shi et al., 2026), Twin Rollouts (Ma et al., 2026), Shared Noise, and Posterior Alignment, covering independent prediction and alternative cross-branch coupling strategies. All methods use the same action-conditioned DiT backbone with matched training groups, actions, optimization budgets, and evaluation protocols. We use a frozen Wan2.1 VAE (Team Wan et al., 2025), initialize from ACWM-DiT, and use $K = 3$ branches by default. The mechanism interpreter and prior are trained first and then frozen. Full benchmark construction, architecture, optimization, and sampling details are **in Appendix E.1 and Appendix D**.

Evaluation metrics. We evaluate both single-rollout prediction quality and shared-world consistency. For single-rollout prediction, we use Masked-MSE (M-MSE) from ACWM-Phys (Xue et al., 2026), which emphasizes temporally changing, action-affected regions. For shared-world evaluation, a frozen state extractor maps each generated video to an observable trajectory $\hat{S}^{(k)}$. Replaying the corresponding action from the same initial state under candidate simulator parameters w gives $S^{(k)}(w)$. Using a normalized environment-specific state distance d, we define

$$\ell_k(w) = \frac{1}{T}\sum_{t=1}^{T} d\Big(\hat{S}_t^{(k)}, S_t^{(k)}(w)\Big), \tag{10}$$

$$E_{\text{ind}} = \frac{1}{K}\sum_{k=1}^{K} \min_{w_k} \ell_k(w_k), \tag{11}$$

$$E_{\text{shared}} = \min_{w} \frac{1}{K}\sum_{k=1}^{K} \ell_k(w), \qquad \Delta_{\text{world}} = E_{\text{shared}} - E_{\text{ind}}. \tag{12}$$

Here E_{ind} measures branch-wise physical fit, E_{shared} requires one physical configuration to explain all branches jointly, and Δ_{world} measures the additional cost of enforcing a common world. We therefore report all three metrics jointly to distinguish shared-world consistency from overall physical fit quality. The simulator-based evaluator is independent of the learned compatibility score; further details are provided **in Appendix E**.

4.2 Main Results

Table 1 compares OneWorld with independent prediction and alternative cross-branch coupling strategies. We jointly report single-rollout prediction quality, individual physical fit, and shared-world consistency.

Overall performance. OneWorld consistently improves shared-world consistency across all three dynamics regimes. Relative to ACWM-DiT, the mean Δ_{world} decreases from 0.02382 to 0.00374, corresponding to an **84.3%** reduction. At the same time, E_{ind} remains low and E_{shared} decreases substantially, indicating that the smaller gap does not arise from uniformly poor physical fits. OneWorld also achieves the lowest M-MSE in all three environments, improving the mean by approximately 7.7×10^{-4} over ACWM-DiT. Sensitivity analysis on Push Cube shows that increasing K from 2 to 5 raises Δ_{world} with only small M-MSE changes, while λ_{shared} and $\lambda_{\max}$ control the trade-off between consistency and prediction quality. Increasing the number of importance samples stabilizes the estimate, with limited gains beyond $M = 64$ (Figure 4). Additional analyses of ground-truth physical-configuration recovery, interpolation to unseen physical values, and detailed sensitivity trends are provided **in Appendix E.8, Appendix E.9, and Appendix D.6**.

Comparison with alternative coupling strategies. The baselines show that improving action-conditioned prediction alone does not eliminate shared-world inconsistency. Vid2World, CoCo,

Table 1: Multi-intervention comparison: individual versus shared physical fits (E_{ind}, E_{shared}) and their gap Δ_{world}. Values are mean $\pm$ standard deviation over three independent runs; lower is better for all metrics.

Category	Environment	Method	Prediction	Physical fit and consistency		
			M-MSE ↓	E_{ind} ↓	E_{shared} ↓	Δ_{world} ↓
Rigid body	Push Cube	ACWM-DiT	0.03477 ± 0.00034	0.07535 ± 0.00112	0.09788 ± 0.00173	0.02253 ± 0.00109
		Vid2World	0.03428 ± 0.00029	0.07488 ± 0.00096	0.09346 ± 0.00161	0.01858 ± 0.00094
		CoCo	0.03416 ± 0.00031	0.07452 ± 0.00088	0.08891 ± 0.00139	0.01439 ± 0.00082
		Twin Rollouts	0.03435 ± 0.00027	0.07497 ± 0.00104	0.09078 ± 0.00148	0.01581 ± 0.00087
		Shared Noise	0.03464 ± 0.00036	0.08073 ± 0.00121	0.11122 ± 0.00196	0.03049 ± 0.00128
		Posterior Alignment	0.03486 ± 0.00033	0.07547 ± 0.00091	0.08836 ± 0.00132	0.01289 ± 0.00076
		OneWorld	$\mathbf{0.03396 \pm 0.00024}$	$\mathbf{0.07384 \pm 0.00079}$	$\mathbf{0.07741 \pm 0.00108}$	$\mathbf{0.00357 \pm 0.00038}$
Deformable	Push Rope	ACWM-DiT	0.00817 ± 0.00011	0.04128 ± 0.00062	0.05517 ± 0.00091	0.01389 ± 0.00057
		Vid2World	0.00798 ± 0.00009	0.04096 ± 0.00055	0.05302 ± 0.00083	0.01206 ± 0.00049
		CoCo	0.00791 ± 0.00008	0.04072 ± 0.00051	0.05088 ± 0.00076	0.01016 ± 0.00044
		Twin Rollouts	0.00802 ± 0.00010	0.04105 ± 0.00058	0.05214 ± 0.00079	0.01109 ± 0.00047
		Shared Noise	0.00809 ± 0.00012	0.04311 ± 0.00067	0.05972 ± 0.00098	0.01661 ± 0.00063
		Posterior Alignment	0.00823 ± 0.00009	0.04147 ± 0.00053	0.04961 ± 0.00071	0.00814 ± 0.00041
		OneWorld	$\mathbf{0.00771 \pm 0.00007}$	$\mathbf{0.04012 \pm 0.00046}$	$\mathbf{0.04221 \pm 0.00064}$	$\mathbf{0.00209 \pm 0.00026}$
Particle	Pour Water	ACWM-DiT	0.03624 ± 0.00039	0.09037 ± 0.00134	0.12541 ± 0.00217	0.03504 ± 0.00146
		Vid2World	0.03584 ± 0.00035	0.08993 ± 0.00121	0.12072 ± 0.00198	0.03079 ± 0.00132
		CoCo	0.03567 ± 0.00032	0.08951 ± 0.00116	0.11564 ± 0.00184	0.02613 ± 0.00118
		Twin Rollouts	0.03596 ± 0.00036	0.09005 ± 0.00127	0.11822 ± 0.00191	0.02817 ± 0.00124
		Shared Noise	0.03605 ± 0.00041	0.09314 ± 0.00142	0.13291 ± 0.00231	0.03977 ± 0.00157
		Posterior Alignment	0.03635 ± 0.00034	0.09058 ± 0.00119	0.11231 ± 0.00172	0.02173 ± 0.00106
		OneWorld	$\mathbf{0.03521 \pm 0.00028}$	$\mathbf{0.08846 \pm 0.00103}$	$\mathbf{0.09402 \pm 0.00153}$	$\mathbf{0.00556 \pm 0.00061}$

and Twin Rollouts reduce the gap relative to independent ACWM-DiT generation, but remain substantially above OneWorld. Shared Noise slightly improves M-MSE while increasing the shared-world gap, indicating that common randomness is not sufficient to enforce a common physical world. Posterior Alignment provides the strongest alternative on the consistency metrics by using the same mechanism interpreter, yet direct posterior similarity remains markedly weaker than the prior-corrected shared-support criterion used by OneWorld.

4.3 Qualitative Analysis on RoboDesk



Figure 3: **Different actions from the same initial world on RoboDesk.** Branches share the initial state and hidden physics but differ in action. Consistency concerns a common physical explanation after accounting for contact and release conditions, rather than identical displacements.

Figure 3 compares action responses from the same initial world in two RoboDesk cases (Kannan et al., 2021). OneWorld preserves distinct outcomes for slow versus fast drawer pulls and short

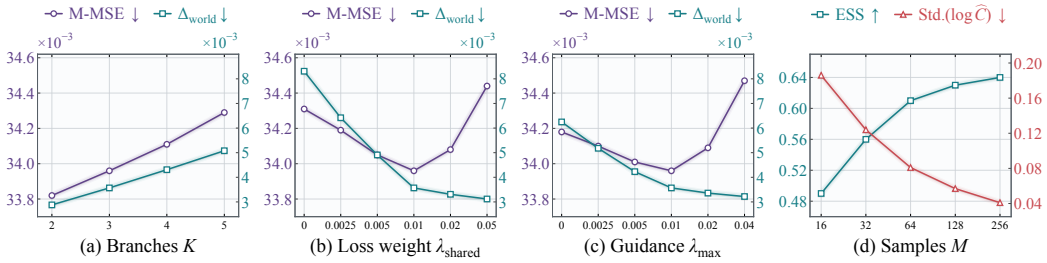


Figure 4: Sensitivity of OneWorld to the number of intervention branches K, shared-loss weight λ_{shared}, maximum guidance strength $\lambda_{\max}$, and the number of importance samples M on Push Cube.

versus long door pushes while maintaining coherent post-contact dynamics across interventions. **Appendix C.4** provides the complete setup and contact/release-conditioned interpretation.

4.4 Ablation Studies

Table 2 isolates three components on **Push Cube** with the same backbone and evaluation protocol.

Table 2: Ablation of OneWorld. "Mismatch Neg." denotes mismatched-world supervision for the mechanism interpreter. "Shared Train" denotes shared-world flow training with $\mathcal{L}_{\text{shared}}$, and "Flow Guide" denotes shared-world flow guidance during sampling.

Method	**Components**			**Metrics**		
	Mismatch Neg.	**Shared Train**	**Flow Guide**	**M-MSE ↓**	E_{shared} ↓	Δ_{world} ↓
ACWM-DiT	–	–	–	0.03477	0.09788	0.02253
No Mismatch Neg.	✗	✓	✓	0.03442	0.08495	0.01084
No Shared Train	✓	✗	✓	0.03431	0.08260	0.00831
No Flow Guide	✓	✓	✗	0.03418	0.08024	0.00625
OneWorld	✓	✓	✓	**0.03396**	**0.07741**	**0.00357**

Removing mismatched-world supervision worsens the gap, supporting its role in learning compatibility without establishing physical semantics for individual latent coordinates. Shared-world training and guidance each reduce the gap on their own; together they give the lowest gap and E_{shared} among tested variants. M-MSE remains close across variants (0.03396–0.03477).

5 Conclusion

We introduced OneWorld to improve shared-world consistency among action-conditioned futures. A physical mechanism interpreter and prior-corrected shared-world evidence jointly couple multiple intervention branches during training and sampling. Our simulator-based evaluation separates individual physical fit from the additional error introduced by enforcing one shared physical configuration. Across three controlled environments, OneWorld consistently improves cross-intervention physical consistency while maintaining strong single-rollout prediction quality. Recovery and interpolation analyses further support the learned shared physical explanations beyond the discrete training configurations. These results highlight shared-world consistency as an important complement to conventional single-rollout evaluation for action-conditioned world models.

AI USE STATEMENT

Generative AI tools were used for language editing, manuscript organization, and LaTeX drafting assistance. All AI-assisted text, equations, citations, experimental descriptions, and numerical claims were reviewed by the authors. The authors take responsibility for the correctness and originality of the final submission.

ETHICS STATEMENT

This work studies the consistency of action-conditioned world models in controlled simulation and on public video-prediction benchmarks. It does not involve human subjects, personally identifying information, or deployment in a live robotic system. The simulator-based evaluation is intended to diagnose model inconsistency rather than to infer physical properties of real individuals or private environments.

REPRODUCIBILITY STATEMENT

Section 3 specifies the shared-world evidence score, mechanism learning objective, flow-training objective, and sampling-time guidance. Section 4.1 describes the main evaluation protocol. Appendix D provides architecture, optimization, and sampling details, while Appendix E specifies environment construction, group formation, state extraction, physical fitting, grid-density checks, physical-configuration recovery, and interpolation evaluation. We will release the reconstructed environments, training configurations, evaluator code, and split definitions with the source code.

REFERENCES

Pulkit Agrawal, Ashvin V. Nair, Pieter Abbeel, Jitendra Malik, and Sergey Levine. Learning to poke by poking: Experiential learning of intuitive physics. In *Advances in Neural Information Processing Systems (NIPS)*, volume 29, 2016.

Ossama Ahmed, Frederik Träuble, Anirudh Goyal, Alexander Neitz, Manuel Wüthrich, Yoshua Bengio, Bernhard Schölkopf, and Stefan Bauer. CausalWorld: A robotic manipulation benchmark for causal structure and transfer learning. In *International Conference on Learning Representations (ICLR)*, 2021.

Mohammad Babaeizadeh, Mohammad Taghi Saffar, Suraj Nair, Sergey Levine, Chelsea Finn, and Dumitru Erhan. FitVid: Overfitting in pixel-level video prediction. *arXiv preprint arXiv:2106.13195*, 2021.

Anton Bakhtin, Laurens van der Maaten, Justin Johnson, Laura Gustafson, and Ross Girshick. PHYRE: A new benchmark for physical reasoning. In *Advances in Neural Information Processing Systems (NeurIPS)*, volume 32, 2019.

Hritik Bansal, Zongyu Lin, Tianyi Xie, Zeshun Zong, Michal Yarom, Yonatan Bitton, Chenfanfu Jiang, Yizhou Sun, Kai-Wei Chang, and Aditya Grover. VideoPhy: Evaluating physical commonsense for video generation. In *International Conference on Learning Representations (ICLR)*, 2025.

Hritik Bansal, Clark Peng, Yonatan Bitton, Roman Goldenberg, Aditya Grover, and Kai-Wei Chang. VideoPhy-2: A challenging action-centric physical commonsense evaluation in video generation. In *International Conference on Learning Representations (ICLR)*, 2026.

Fabien Baradel, Natalia Neverova, Julien Mille, Greg Mori, and Christian Wolf. CoPhy: Counterfactual learning of physical dynamics. In *International Conference on Learning Representations (ICLR)*, 2020.

Peter Battaglia, Razvan Pascanu, Matthew Lai, Danilo Jimenez Rezende, and Koray Kavukcuoglu. Interaction networks for learning about objects, relations and physics. In *Advances in Neural Information Processing Systems (NIPS)*, volume 29, 2016.

Jake Bruce, Michael D. Dennis, Ashley Edwards, Jack Parker-Holder, Yuge Shi, Edward Hughes, Matthew Lai, Aditi Mavalankar, Richie Steigerwald, Chris Apps, Yusuf Aytar, Sarah Maria Elisabeth Bechtle, Feryal Behbahani, Stephanie C. Y. Chan, Nicolas Heess, Lucy Gonzalez, Simon Osindero, Sherjil Ozair, Scott Reed, Jingwei Zhang, Konrad Zolna, Jeff Clune, Nando De Freitas, Satinder Singh, and Tim Rocktäschel. Genie: Generative interactive environments. In *Proceedings of the 41st International Conference on Machine Learning (ICML)*, pp. 4603–4623, 2024.

João Carreira and Andrew Zisserman. Quo vadis, action recognition? a new model and the kinetics dataset. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 6299–6308, 2017.

Sudeep Dasari, Frederik Ebert, Stephen Tian, Suraj Nair, Bernadette Bucher, Karl Schmeckpeper, Siddharth Singh, Sergey Levine, and Chelsea Finn. RoboNet: Large-scale multi-robot learning. In *Proceedings of the Conference on Robot Learning (CoRL)*, pp. 885–897, 2020.

Emily Denton and Rob Fergus. Stochastic video generation with a learned prior. In *Proceedings of the 35th International Conference on Machine Learning (ICML)*, pp. 1174–1183, 2018.

Chelsea Finn and Sergey Levine. Deep visual foresight for planning robot motion. In *IEEE International Conference on Robotics and Automation (ICRA)*, pp. 2786–2793, 2017.

Xingzhuo Guo, Yu Zhang, Baixu Chen, Haoran Xu, Jianmin Wang, and Mingsheng Long. Dynamical diffusion: Learning temporal dynamics with diffusion models. In *International Conference on Learning Representations (ICLR)*, 2025.

Yanjiang Guo, Lucy Xiaoyang Shi, Jianyu Chen, and Chelsea Finn. Ctrl-World: A controllable generative world model for robot manipulation. In *International Conference on Learning Representations (ICLR)*, 2026.

Agrim Gupta, Stephen Tian, Yunzhi Zhang, Jiajun Wu, Roberto Martin-Martin, and Li Fei-Fei. MaskViT: Masked visual pre-training for video prediction. In *International Conference on Learning Representations (ICLR)*, 2023.

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

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

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

Danijar Hafner, Jurgis Pasukonis, Jimmy Ba, and Timothy Lillicrap. Mastering diverse control tasks through world models. *Nature*, 640:647–653, 2025.

Nicklas A. Hansen, Hao Su, and Xiaolong Wang. Temporal difference learning for model predictive control. In *Proceedings of the 39th International Conference on Machine Learning (ICML)*, pp. 8387–8406, 2022.

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

Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 770–778, 2016.

Jonathan Ho, Ajay Jain, and Pieter Abbeel. Denoising diffusion probabilistic models. In *Advances in Neural Information Processing Systems (NeurIPS)*, volume 33, pp. 6840–6851, 2020.

Jonathan Ho, Tim Salimans, Alexey Gritsenko, William Chan, Mohammad Norouzi, and David J. Fleet. Video diffusion models. In *Advances in Neural Information Processing Systems (NeurIPS)*, volume 35, 2022.

Siqiao Huang, Jialong Wu, Qixing Zhou, Shangchen Miao, and Mingsheng Long. Vid2World: Crafting video diffusion models to interactive world models. In *International Conference on Learning Representations (ICLR)*, 2026.

Ziqi Huang, Yinan He, Jiashuo Yu, Fan Zhang, Chenyang Si, Yuming Jiang, Yuanhan Zhang, Tianxing Wu, Qingyang Jin, Nattapol Chanpaisit, Yaohui Wang, Xinyuan Chen, Limin Wang, Dahua Lin, Yu Qiao, and Ziwei Liu. VBench: Comprehensive benchmark suite for video generative models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 21807–21818, 2024.

Lukasz Kaiser, Mohammad Babaeizadeh, Piotr Milos, Blazej Osinski, Roy H. Campbell, Konrad Czechowski, Dumitru Erhan, Chelsea Finn, Piotr Kozakowski, Sergey Levine, Afroz Mohiuddin, Ryan Sepassi, George Tucker, and Henryk Michalewski. Model based reinforcement learning for atari. In *International Conference on Learning Representations (ICLR)*, 2020.

Harini Kannan, Danijar Hafner, Chelsea Finn, and Dumitru Erhan. RoboDesk: A multi-task reinforcement learning benchmark. RoboDesk Benchmark, 2021.

Dacheng Li, Yunhao Fang, Yukang Chen, Shuo Yang, Shiyi Cao, Justin Wong, Michael Luo, Xiaolong Wang, Hongxu Yin, Joseph E. Gonzalez, Ion Stoica, Song Han, and Yao Lu. WorldModelBench: Judging video generation models as world models. In *Advances in Neural Information Processing Systems (NeurIPS), Datasets and Benchmarks Track*, volume 38, 2025.

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

Yaron Lipman, Ricky T. Q. Chen, Heli Ben-Hamu, Maximilian Nickel, and Matthew Le. Flow matching for generative modeling. In *International Conference on Learning Representations (ICLR)*, 2023.

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

Yu Ma, Hongli Shi, and Xinran Xu. Twin rollouts: Noise-coupled counterfactual branching in interactive video world models. *arXiv preprint arXiv:2608.08982*, 2026.

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

Robert Osazuwa Ness, Ricardo Cannizzaro, Yunshu Wu, and Lars Kunze. Multiverse mechanica: A testbed for learning game mechanics via counterfactual worlds. In *International Conference on Learning Representations (ICLR)*, 2026.

William Peebles and Saining Xie. Scalable diffusion models with transformers. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 4195–4205, 2023.

Tobias Pfaff, Meire Fortunato, Alvaro Sanchez-Gonzalez, and Peter W. Battaglia. Learning mesh-based simulation with graph networks. In *International Conference on Learning Representations (ICLR)*, 2021.

Robin Rombach, Andreas Blattmann, Dominik Lorenz, Patrick Esser, and Björn Ommer. High-resolution image synthesis with latent diffusion models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 10684–10695, 2022.

Olaf Ronneberger, Philipp Fischer, and Thomas Brox. U-Net: Convolutional networks for biomedical image segmentation. In *Medical Image Computing and Computer-Assisted Intervention (MICCAI)*, pp. 234–241, 2015.

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

Julian Schrittwieser, Ioannis Antonoglou, Thomas Hubert, Karen Simonyan, Laurent Sifre, Simon Schmitt, Arthur Guez, Edward Lockhart, Demis Hassabis, Thore Graepel, Timothy Lillicrap, and David Silver. Mastering atari, go, chess and shogi by planning with a learned model. *Nature*, 588: 604–609, 2020.

Yuhong Shi, Zhenhao Chu, Jie Wei, Jun Hao, Jianyi Liu, and Jingwen Fu. Overcoming statistical bias in action-controllable world models. *arXiv preprint arXiv:2608.04653*, 2026.

Jiaming Song, Chenlin Meng, and Stefano Ermon. Denoising diffusion implicit models. In *International Conference on Learning Representations (ICLR)*, 2021a.

Yang Song, Jascha Sohl-Dickstein, Diederik P. Kingma, Abhishek Kumar, Stefano Ermon, and Ben Poole. Score-based generative modeling through stochastic differential equations. In *International Conference on Learning Representations (ICLR)*, 2021b.

Team Wan, Ang Wang, Baole Ai, Bin Wen, Chaojie Mao, Chen-Wei Xie, Di Chen, Feiwu Yu, Haiming Zhao, Jianxiao Yang, Jianyuan Zeng, Jiayu Wang, Jingfeng Zhang, Jingren Zhou, Jinkai Wang, Jixuan Chen, Kai Zhu, Kang Zhao, Keyu Yan, Lianghua Huang, et al. Wan: Open and advanced large-scale video generative models. *arXiv preprint arXiv:2503.20314*, 2025.

Thomas Unterthiner, Sjoerd van Steenkiste, Karol Kurach, Raphaël Marinier, Marcin Michalski, and Sylvain Gelly. Towards accurate generative models of video: A new metric & challenges. *arXiv preprint arXiv:1812.01717*, 2018.

Vikram Voleti, Alexia Jolicoeur-Martineau, and Christopher Pal. MCVD: Masked conditional video diffusion for prediction, generation, and interpolation. In *Advances in Neural Information Processing Systems (NeurIPS)*, volume 35, 2022.

Jing Wang, Ao Ma, Ke Cao, Jun Zheng, Jiasong Feng, Zhanjie Zhang, Wanyuan Pang, and Xiaodan Liang. WISA: World simulator assistant for physics-aware text-to-video generation. In *Advances in Neural Information Processing Systems (NeurIPS)*, volume 38, 2025a.

Sen Wang, Jingyi Tian, Le Wang, Zhimin Liao, Jiayi Li, Huaiyi Dong, Kun Xia, Sanping Zhou, Wei Tang, and Gang Hua. SAMPO: Scale-wise autoregression with motion prompt for generative world models. In *Advances in Neural Information Processing Systems (NeurIPS)*, volume 38, 2025b.

Zhou Wang, Alan C. Bovik, Hamid R. Sheikh, and Eero P. Simoncelli. Image quality assessment: From error visibility to structural similarity. *IEEE Transactions on Image Processing (TIP)*, 13 (4):600–612, 2004.

Zijun Wang, Panwen Hu, Jing Wang, Terry Jingchen Zhang, Yuhao Cheng, Long Chen, Yiqiang Yan, Zutao Jiang, Hanhui Li, and Xiaodan Liang. ProPhy: Progressive physical alignment for dynamic world simulation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2026.

Bohan Wu, Suraj Nair, Roberto Martin-Martin, Li Fei-Fei, and Chelsea Finn. Greedy hierarchical variational autoencoders for large-scale video prediction. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 2318–2328, 2021.

Jiajun Wu, Ilker Yildirim, Joseph J. Lim, William T. Freeman, and Joshua B. Tenenbaum. Galileo: Perceiving physical object properties by integrating a physics engine with deep learning. In *Advances in Neural Information Processing Systems (NIPS)*, volume 28, 2015.

Jialong Wu, Shaofeng Yin, Ningya Feng, Xu He, Dong Li, Jianye Hao, and Mingsheng Long. iVideoGPT: Interactive videogpts are scalable world models. In *Advances in Neural Information Processing Systems (NeurIPS)*, volume 37, 2024.

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

Mengyue Yang, Furui Liu, Zhitang Chen, Xinwei Shen, Jianye Hao, and Jun Wang. CausalVAE: Disentangled representation learning via neural structural causal models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 9593–9602, 2021.

Sherry Yang, Yilun Du, Seyed Ghasemipour, Jonathan Tompson, Leslie Kaelbling, Dale Schuurmans, and Pieter Abbeel. Learning interactive real-world simulators. In *International Conference on Learning Representations (ICLR)*, 2024.

Kexin Yi, Chuang Gan, Yunzhu Li, Pushmeet Kohli, Jiajun Wu, Antonio Torralba, and Joshua B. Tenenbaum. CLEVRER: Collision events for video representation and reasoning. In *International Conference on Learning Representations (ICLR)*, 2020.

Lijun Yu, Yong Cheng, Kihyuk Sohn, Jose Lezama, Han Zhang, Huiwen Chang, Alexander G. Hauptmann, Ming-Hsuan Yang, Yuan Hao, Irfan Essa, and Lu Jiang. MAGVIT: Masked generative video transformer. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 10459–10469, 2023.

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. In *International Conference on Learning Representations (ICLR)*, 2026.

Richard Zhang, Phillip Isola, Alexei A. Efros, Eli Shechtman, and Oliver Wang. 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.

A Properties of Shared-World Evidence

This appendix provides additional details on the shared-world evidence defined in Eq. 4. We first derive its prior-corrected evidence-ratio interpretation under an idealized Bayesian setting and then establish the neutrality of a branch whose posterior contains no information beyond the initial-scene prior. These results motivate the prior-corrected form of the compatibility score used by OneWorld.

A.1 An Idealized Evidence-Ratio Interpretation

Fix the initial observation, intervention sequences, and flow time, and write x_k for the outcome representation associated with branch k. We suppress this fixed conditioning to simplify notation.

Let $\pi(W)$ be a normalized prior over the latent mechanism and let

$$L_k(W) = p(x_k \mid W)$$

denote the likelihood of the outcome of branch k under its corresponding fixed action. Define the marginal likelihood

$$Z_k = \int \pi(W) L_k(W)\, dW,$$

which is assumed to be finite and positive.

Under an idealized Bayesian model, the exact branch posterior is

$$q_k(W) = \frac{\pi(W) L_k(W)}{Z_k}.$$

Substituting the branch posteriors into the shared-world evidence yields

$$\begin{aligned}
\mathcal{C} &= \int \pi(W)^{1-K} \prod_{k=1}^{K} q_k(W)\, dW \\
&= \int \pi(W)^{1-K} \prod_{k=1}^{K} \frac{\pi(W) L_k(W)}{Z_k}\, dW \\
&= \frac{\int \pi(W) \prod_{k=1}^{K} L_k(W)\, dW}{\prod_{k=1}^{K} Z_k}.
\end{aligned}$$

The numerator is the marginal likelihood under a model in which all branches share a single realization of W. The denominator is the product of the branch-wise marginal likelihoods and corresponds to independently drawing one mechanism from the same prior for each branch.

The correction factor

$$\pi(W)^{1-K}$$

therefore removes the repeated prior factors introduced by multiplying the K posterior distributions. The shared-mechanism model contains the prior only once.

Under these assumptions, a larger value of $\mathcal{C}$ represents stronger relative evidence that the observed action–outcome branches can be jointly explained by a common mechanism. In OneWorld, q_ϕ is learned from relational supervision rather than exact likelihoods. We therefore use $\mathcal{C}_\tau$ as a differentiable compatibility score and do not interpret its absolute magnitude as a calibrated Bayes factor.

A.2 Neutrality of a Prior-Matching Branch

For a set of intervention branches $\mathcal{S}$, define

$$\mathcal{C}_\mathcal{S} = \int \pi_\phi(W \mid I_0)^{1-|\mathcal{S}|} \prod_{k\in\mathcal{S}} q_\phi^{(k)}(W)\, dW.$$

Assume the integral is finite and the prior has positive density on the relevant support. Consider a branch $j \in \mathcal{S}$ whose posterior contains no additional information beyond the initial observation:

$$q_\phi^{(j)}(W) = \pi_\phi(W \mid I_0).$$

Then

$$\begin{aligned}
\mathcal{C}_\mathcal{S} &= \int \pi_\phi(W \mid I_0)^{1-|\mathcal{S}|} q_\phi^{(j)}(W) \prod_{k\in\mathcal{S}\setminus\{j\}} q_\phi^{(k)}(W)\, dW \\
&= \int \pi_\phi(W \mid I_0)^{2-|\mathcal{S}|} \prod_{k\in\mathcal{S}\setminus\{j\}} q_\phi^{(k)}(W)\, dW \\
&= \mathcal{C}_{\mathcal{S}\setminus\{j\}}.
\end{aligned}$$

Thus, adding a branch whose posterior equals the prior does not change the shared-world evidence. For a single branch, the score is 1 by normalization, and it also remains 1 when all branch posteriors equal the prior.

This property is algebraic. It does not by itself establish that the learned interpreter produces a prior-matching posterior for every empirically uninformative intervention. The latter remains a property of the learned representation rather than of the score definition.

B External Action-Conditioned Video Prediction

We additionally evaluate the action-conditioned DiT backbone used by OneWorld on BAIR (Finn & Levine, 2017) and RoboNet (Dasari et al., 2020) using standard video-prediction metrics. These datasets provide real robot interaction trajectories but not the controlled multi-intervention physical configurations required by the shared-world evaluator, so this appendix focuses on single-rollout prediction quality.

B.1 Evaluation Scope

For both datasets, the external prediction task has the form

$$p\left(V_{\text{future}} \mid V_{\text{context}}, A_{\text{future}}\right),$$

where the model observes a short visual context together with the future robot action sequence and predicts the corresponding future video.

For BAIR, we use two observed context frames and predict the following twelve frames. For RoboNet, we use two observed context frames and predict the following ten frames. All models compared within each dataset use the same context length and prediction horizon.

Both datasets are evaluated at a spatial resolution of 64×64.

BAIR and RoboNet are evaluated with conventional single-rollout metrics; the shared-world metrics are reserved for the controlled environments in the main experiments.

B.2 Preprocessing and Temporal Alignment

Frames are resized to 64×64 before video encoding. The same deterministic spatial transformation is applied to all frames from one trajectory so that preprocessing does not introduce artificial temporal motion.

RGB values are normalized using the preprocessing convention of the frozen video autoencoder. No random crop, horizontal flip, color augmentation, or other test-time augmentation is applied during evaluation.

Robot actions are temporally aligned with their corresponding target transitions after frame sampling. Each action vector is standardized using the mean and standard deviation computed from the training split of the corresponding dataset.

For BAIR, evaluation uses 2,048 held-out prediction clips. For RoboNet, evaluation uses 2,048 held-out clips sampled from trajectories that do not appear in the training split. The same evaluation clips are reused across all locally evaluated models.

Table 3: Protocol for external action-conditioned video prediction.

Dataset	**Resolution**	**Context**	**Prediction**	**Evaluation clips**
BAIR	64×64	2 frames	12 frames	2,048
RoboNet	64×64	2 frames	10 frames	2,048

B.3 Model Configuration on BAIR and RoboNet

The external experiments use the same action-conditioned latent DiT family as the main model. A dataset-specific action-conditioned DiT is trained for each benchmark with the frozen Wan2.1 causal VAE (Team Wan et al., 2025). Because BAIR and RoboNet do not provide the grouped physical supervision used by OneWorld, training is restricted to standard branch-wise action-conditioned flow matching; the mechanism interpreter, shared-world loss, and shared-world guidance are not used in this appendix.

Table 4: Training configuration for the external video prediction benchmarks.

Setting	**BAIR**	**RoboNet**
Backbone	Action-conditioned DiT	Action-conditioned DiT
Video representation	Frozen Wan2.1 causal VAE	Frozen Wan2.1 causal VAE
Optimizer	AdamW	AdamW
Learning rate	1×10^{-5}	1×10^{-5}
Batch size	8	8
Training updates	40,000	50,000
Inference steps	50	50
Mechanism interpreter	Not used	Not used
Shared-world loss	Not used	Not used
Shared-world guidance	Not used	Not used

B.4 Baselines and Result Provenance

The comparison includes recurrent video predictors, hierarchical latent variable models, masked-token predictors, autoregressive world models, and diffusion-based approaches.

Specifically, we compare against SVG (Denton & Fergus, 2018), GHVAE (Wu et al., 2021), FitVid (Babaeizadeh et al., 2021), MaskViT (Gupta et al., 2023), iVideoGPT (Wu et al., 2024), DPM, DyDiff (Guo et al., 2025), and SAMPO (Wang et al., 2025b) when the corresponding metric is available under the matched action-conditioned protocol.

Baseline values are taken from the corresponding published results or from the matched-protocol comparison reported by the later method when the original paper does not contain the complete metric set. We do not combine results reported at different spatial resolutions in the same table. The DPM values are the diffusion-baseline results reported together with DyDiff, which is indicated by the starred venue entry.

Our action-conditioned DiT backbone is evaluated using the protocol described above. A dash indicates that a metric was not available under a sufficiently matched action-conditioned setting.

B.5 Metric Computation

We report Fréchet Video Distance (FVD) (Unterthiner et al., 2018), peak signal-to-noise ratio (PSNR), structural similarity (SSIM) (Wang et al., 2004), and LPIPS (Zhang et al., 2018).

FVD is computed using I3D video features (Carreira & Zisserman, 2017) extracted from 2,048 generated clips and the corresponding 2,048 reference clips. Generated and reference videos use identical spatial preprocessing and the same temporal horizon within each dataset.

PSNR, SSIM, and LPIPS are evaluated over predicted future frames only. Frame-level scores are first averaged temporally within each predicted sequence and are then averaged across the evaluation set.

Following the reporting convention of the compared methods, SSIM and LPIPS values in Table 5 are multiplied by 100.

B.6 External Prediction Results

Table 5: Comparison with existing action-conditioned video prediction methods on BAIR and RoboNet at 64×64 resolution. Lower is better for FVD and LPIPS; higher is better for PSNR and SSIM. SSIM and LPIPS are multiplied by 100. *DPM denotes the diffusion baseline reported with DyDiff.

Dataset	Method	Venue	FVD ↓	PSNR ↑	SSIM ↑	LPIPS ↓
BAIR	MaskViT	ICLR'23	70.5	–	–	–
	iVideoGPT	NeurIPS'24	60.8	24.5	90.2	5.0
	DPM	ICLR'25*	48.5	25.9	92.0	4.5
	DyDiff	ICLR'25	<u>45.0</u>	26.2	92.4	4.2
	SAMPO	NeurIPS'25	55.5	<u>26.7</u>	<u>94.7</u>	<u>3.7</u>
	Our DiT backbone	**Ours**	**41.2**	**27.9**	**95.8**	**2.8**
RoboNet	MaskViT	ICLR'23	133.5	23.2	80.5	4.2
	SVG	NeurIPS'19	123.2	23.9	87.8	6.0
	GHVAE	CVPR'21	95.2	24.7	89.1	3.6
	FitVid	arXiv'21	62.5	28.2	89.3	<u>2.4</u>
	iVideoGPT	NeurIPS'24	63.2	27.8	90.6	4.9
	DPM	ICLR'25*	77.0	26.4	87.3	6.0
	DyDiff	ICLR'25	67.7	26.5	87.5	5.9
	SAMPO	NeurIPS'25	<u>57.1</u>	<u>29.3</u>	<u>94.1</u>	3.3
	Our DiT backbone	**Ours**	**52.6**	**30.6**	**95.3**	**1.7**

Results on BAIR. Our DiT backbone obtains an FVD of 41.2, compared with 45.0 for DyDiff. It also obtains a PSNR of 27.9, an SSIM of 95.8, and an LPIPS of 2.8 under the matched evaluation protocol.

These results show that the action-conditioned backbone used by our framework maintains strong conventional single-rollout prediction quality.

Results on RoboNet. Our DiT backbone obtains an FVD of 52.6, a PSNR of 30.6, an SSIM of 95.3, and an LPIPS of 1.7.

Together with the BAIR results, this confirms that the underlying action-conditioned DiT backbone remains competitive under standard video-prediction metrics.

C RoboDesk Qualitative Evaluation Setup

The RoboDesk experiments (Kannan et al., 2021) in Figure 3 are designed as a controlled qualitative transfer study rather than as an additional single-rollout benchmark. We use two articulated manipulation tasks, *Open Drawer* and *Sliding Door*, because both expose a clear separation between the commanded action and latent interaction properties. For each task, the rendered initial observation fixes the robot pose, object geometry, camera, and articulated-joint state, while friction and damping are kept hidden from the video model.

C.1 Environment and Physical Configurations

For **Open Drawer**, the hidden physical configuration consists of the drawer-rail friction coefficient and viscous joint damping. We use three levels of rail friction, $\{0.18, 0.39, 0.67\}$, and three levels of damping, $\{0.06, 0.15, 0.29\}$ in simulator-native units, yielding nine physical configurations. The action family contains slow pull, fast pull, and pull–release sequences. The first two are visualized in Figure 3; the third is used during grouped training to provide an additional intervention with a different contact and release profile.

For **Sliding Door**, the hidden physical configuration consists of the track friction coefficient and guide damping. The three friction levels are $\{0.11, 0.28, 0.52\}$ and the three damping levels are $\{0.04, 0.12, 0.23\}$, again producing nine configurations. The action family contains short right push, long right push, and right-push–reverse sequences. The first two are shown in the qualitative comparison. The reverse sequence is included during grouped training because it provides a stronger diagnostic of inertia and sliding resistance than displacement alone.

All trajectories contain 37 frames rendered at 240×240 resolution and 10 Hz, matching the temporal and spatial format used by the controlled experiments in the main paper. For each task, we use 192/48/48 initial states for the train/validation/test splits. Each initial state contributes one same-world group and one mismatched-world group, giving 384/96/96 counterfactual groups.

Table 6: RoboDesk setup used for the qualitative comparison.

Task	Initial states	Groups	K	Hidden physical factors	Configs.
Open Drawer	192/48/48	384/96/96	3	rail friction, joint damping	3×3
Sliding Door	192/48/48	384/96/96	3	track friction, guide damping	3×3

C.2 Same-World and Mismatched-World Groups

The mechanism interpreter is trained using the same relational supervision principle as in the main controlled experiments. A same-world group contains three action sequences executed after resetting RoboDesk to the same complete initial simulator state and the same hidden physical configuration. The actions deliberately differ in duration, speed, or direction, so trajectories within one group need not have similar displacements.

A mismatched-world group is formed from the same visible initial state while assigning different hidden configurations to the intervention branches. At least one of the two hidden physical factors differs by one grid level between any two branches in a negative group. Because the camera, robot pose, articulated-object pose, and visible scene are shared, the mechanism-learning label cannot

be inferred from static appearance alone. Positive and negative groups are sampled with equal probability.

For Figure 3, we display two branches from each held-out three-branch group because a two-row visualization makes the action-dependent difference easier to inspect. The third branch remains part of the mechanism learning and shared-world training objective.

C.3 Training Protocol

We train one action-conditioned DiT checkpoint for each RoboDesk task using the corresponding training trajectories. ACWM-DiT uses this checkpoint without cross-branch coupling. OneWorld is initialized from the same checkpoint. The mechanism interpreter and initial-scene prior are first trained on the RoboDesk same-world and mismatched-world groups, after which they are frozen and the world model is optimized using $\mathcal{L}_{\text{flow}} + \lambda_{\text{shared}}\mathcal{L}_{\text{shared}}$.

We use the same mechanism architecture and optimization schedule as in Appendix D: $K = 3$ branches, 20,000 mechanism-learning updates, 12,000 world-model updates, $\lambda_{\text{shared}} = 0.01$, and 50 flow steps. Sampling uses independent Gaussian noise for the different branches and the same $\lambda_\tau = 0.01\tau^2$ guidance schedule used in the main experiments.

For the qualitative baselines, Vid2World and CoCo are instantiated on the same task-specific action-conditioned DiT backbone and trained using the same training trajectories and optimization budget. All compared methods observe the same initial frames and action sequences. No method is given access to the hidden friction or damping values at training or inference time.

C.4 Qualitative Comparison Protocol

The qualitative comparison is constructed from held-out initial states and uses identical action sequences across methods. For each displayed branch, the same Gaussian noise seed is reused across methods, while the two branches within one method retain independent noise. This isolates differences in the learned dynamics from differences caused by sampling initialization.

Open Drawer. Slow and fast pulls can produce different displacements and release velocities. Consistency therefore concerns whether the drawer's subsequent deceleration, sliding, and stopping can be explained by one resistance and sliding-dynamics model, given each branch's action and contact state. For ACWM-DiT, Vid2World, or CoCo, individually plausible and action-responsive predictions could still imply high resistance in one branch and low resistance in the other. Such a conflict must persist after accounting for the different release conditions to constitute shared-world inconsistency. OneWorld seeks to preserve the distinct outcomes of slow and fast pulls while making their physical explanations compatible.

Sliding Door. A short right push and a long right push differ in contact duration, so different door displacements are expected. The informative comparison is the motion after contact ends, conditioned on each branch's release velocity. For example, a short-push branch consistent with a high-friction rail would conflict with a long-push branch that requires very low friction even after its larger release velocity is taken into account. OneWorld aims to retain distinct door trajectories while keeping the implied sliding resistance, inertia, and reversal response compatible across actions.

Example selection. The displayed examples are taken from fixed held-out test states selected before rendering the final comparison, and the same temporal indices are shown for every method.

D Implementation Details

D.1 Physical Mechanism Interpreter

The physical mechanism interpreter is intentionally lightweight relative to the video backbone.

For each intervention branch, we average-pool the estimated clean-future latent $\hat{z}_{1,\tau}^{(k)}$ over spatial dimensions while retaining its temporal sequence. A two-layer projection maps each pooled latent token to a 512-dimensional feature.

The action sequence is encoded by a two-layer MLP followed by temporal average pooling, yielding a 256-dimensional action feature. Flow time τ is represented by a 128-dimensional sinusoidal embedding.

The initial observation I_0 is encoded with the frozen Wan2.1 encoder and projected to a 256-dimensional scene representation.

The scene, action, clean-future, and flow-time features are concatenated and processed by a three-layer MLP with hidden width 1024 and SiLU activations.

The mechanism space is a 32-dimensional diagonal Gaussian with $d_W = 32$. The initial-scene prior is

$$\pi_\phi(W \mid I_0) = \mathcal{N}\left(W; \mu_\pi, \mathrm{diag}(\sigma_\pi^2)\right),$$

where the prior network predicts μ_π and $\log \sigma_\pi$. We clamp $\log \sigma_\pi$ to $[-4, 2]$ before converting it to the prior precision $\lambda_{\pi,d} = \sigma_{\pi,d}^{-2}$ and natural mean parameter $\eta_{\pi,d} = \lambda_{\pi,d}\mu_{\pi,d}$.

For branch k, the interpreter predicts an evidence center $m_{\phi,d}^{(k)}$ and a raw precision increment $\rho_{\phi,d}^{(k)}$. The branch contributes

$$\Delta\lambda_{\phi,d}^{(k)} = \mathrm{softplus}\left(\rho_{\phi,d}^{(k)}\right),$$

and the posterior natural parameters are

$$\lambda_{\phi,d}^{(k)} = \lambda_{\pi,d} + \Delta\lambda_{\phi,d}^{(k)}, \qquad \eta_{\phi,d}^{(k)} = \eta_{\pi,d} + \Delta\lambda_{\phi,d}^{(k)} m_{\phi,d}^{(k)}.$$

The resulting branch posterior is

$$q_\phi^{(k)}(W) = \mathcal{N}\left(W; \mu_\phi^{(k)}, \mathrm{diag}\left[(\sigma_\phi^{(k)})^2\right]\right),$$

with

$$(\sigma_{\phi,d}^{(k)})^2 = \left(\lambda_{\phi,d}^{(k)}\right)^{-1}, \qquad \mu_{\phi,d}^{(k)} = \frac{\eta_{\phi,d}^{(k)}}{\lambda_{\phi,d}^{(k)}}.$$

This parameterization treats an action–outcome branch as additional evidence about the initial world. When an intervention contributes little information, $\Delta\lambda_{\phi,d}^{(k)}$ approaches zero and the branch posterior approaches the initial-scene prior. More informative interventions contribute larger precision increments and produce sharper mechanism estimates.

The same construction gives the prior-corrected Gaussian product a positive combined precision in every latent dimension:

$$\sum_{k=1}^{K} \lambda_{\phi,d}^{(k)} - (K-1)\lambda_{\pi,d} = \lambda_{\pi,d} + \sum_{k=1}^{K} \Delta\lambda_{\phi,d}^{(k)} > 0.$$

Thus the shared-world evidence in Eq. 4 is well-defined under the Gaussian parameterization. The physical mechanism interpreter contains approximately 8.7M trainable parameters.

Table 7: Implementation details for the mechanism model and OneWorld training.

Component	Setting
Latent mechanism W	32-dimensional diagonal Gaussian
Posterior update	Additive natural-parameter evidence, $\Delta\lambda = \mathrm{softplus}(\rho)$
Clean-future projector	2-layer MLP, width 512
Action encoder	2-layer MLP, width 256
Flow-time embedding	128-dimensional sinusoidal embedding
Interpreter fusion network	3-layer MLP, width 1024, SiLU
Prior network	2-layer MLP, width 512
Interpreter optimizer	AdamW, lr 1×10^{-4}, weight decay 0.01
Interpreter batch size	16 counterfactual groups
Interpreter training	20,000 updates, 1:1 same-world/mismatched-world groups
World-model optimizer	AdamW, lr 1×10^{-5}
World-model batch size	1 counterfactual group, $K = 3$ branches
World-model updates	12,000 updates
Shared-loss weight	$\lambda_{\mathrm{shared}} = 0.01$
Sampling	50 flow steps
Maximum guidance strength	0.01
Guidance schedule	$\lambda_\tau = 0.01\tau^2$

D.2 Mechanism-Learning Groups

The mechanism interpreter is trained using relational supervision rather than direct labels for individual physical parameters.

A same-world group contains $K = 3$ branches generated after resetting the environment to the same complete initial state and the same hidden physical configuration. Different actions are then executed from that state. The resulting trajectories may differ substantially, but their physical explanations should remain compatible with one common world.

A mismatched-world group also contains three branches but draws them from different hidden physical configurations. Initial visible states are kept matched so that the group label cannot be predicted simply from scene appearance.

For Push Cube and Push Rope, every initial state contributes one same-world group and one mismatched-world group. The resulting train/validation/test group counts are therefore $256/64/64$ from $128/32/32$ initial states.

Positive and negative groups are sampled in a balanced $1{:}1$ ratio during mechanism learning.

The interpreter is not provided with semantic targets such as the value of drag, elasticity, stiffness, or damping. Its latent coordinates are therefore not assumed to correspond one-to-one to named simulator parameters. Instead, the representation is learned only through the compatibility of action–outcome branches.

D.3 Numerical Evaluation of Shared-World Evidence

We evaluate Eq. 4 using importance sampling in the common latent mechanism space.

For a group containing K branches, the proposal distribution is

$$r(W) = \frac{1}{K+1}\left[\pi_\phi(W \mid I_0) + \sum_{k=1}^{K} q_\phi^{(k)}(W)\right].$$

This proposal places probability mass around the prior as well as every branch posterior.

With samples $W_m \sim r(W)$, the evidence is estimated as

$$\widehat{\mathcal{C}}_\tau = \frac{1}{M}\sum_{m=1}^{M} \frac{\pi_\phi(W_m \mid I_0)^{1-K}\prod_{k=1}^{K} q_\phi^{(k)}(W_m)}{r(W_m)}.$$

We use $M = 64$ samples per counterfactual group. Samples are allocated uniformly across the prior and branch-posterior components of the proposal.

All Gaussian densities and importance weights are evaluated in log space. Products of densities are represented as sums of log densities and sample aggregation is performed using log-sum-exp for numerical stability.

D.4 Gradient Computation

The mechanism interpreter is updated only during the first training stage.

During shared-world flow training, its parameters are frozen. However, gradients from $\mathcal{L}_{\text{shared}}$ are allowed to propagate through its inputs and through $\hat{z}_{1,\tau}^{(k)}$ to the action-conditioned vector field.

Freezing the interpreter therefore fixes the learned compatibility criterion while still allowing the video model to change its predictions in directions that improve shared-world evidence.

During sampling-time guidance, we compute

$$g_\tau^{(k)} = \nabla_{z_\tau^{(k)}} \log \left(\mathcal{C}_\tau + \epsilon \right).$$

The velocity prediction inside the clean-future estimate is detached when this gradient is evaluated. This avoids second-order differentiation through the flow backbone while retaining the direct derivative from the current latent to the compatibility score.

The guided velocity is

$$\widetilde{v}_\theta^{(k)} = v_\theta^{(k)} + \lambda_\tau g_\tau^{(k)},$$

where

$$\lambda_\tau = 0.01\tau^2.$$

The quadratic schedule applies little guidance at early flow times, when the predicted clean future is highly uncertain, and gradually increases the influence of the shared-world score as generation progresses.

D.5 Training Schedule

Training consists of two stages.

Stage 1: mechanism learning. The pretrained action-conditioned video backbone is frozen. The mechanism interpreter and initial-scene prior are optimized for 20,000 updates using Eq. 5.

We use AdamW (Loshchilov & Hutter, 2019) with learning rate 1×10^{-4}, weight decay 0.01, and a batch size of 16 counterfactual groups. Same-world and mismatched-world groups are sampled with equal probability.

At every update, flow time is sampled uniformly from $[0, 1]$ and the clean-future estimate is computed from the frozen ACWM-DiT backbone.

Stage 2: shared-world video-model training. After mechanism learning, both the mechanism interpreter and the initial-scene prior are frozen.

The action-conditioned video model is initialized from the corresponding ACWM-DiT checkpoint and optimized for 12,000 updates with

$$\mathcal{L} = \mathcal{L}_{\text{flow}} + \lambda_{\text{shared}} \mathcal{L}_{\text{shared}},$$

using

$$\lambda_{\text{shared}} = 0.01.$$

We use AdamW with learning rate 1×10^{-5} and one counterfactual group per batch. Unless otherwise specified, each group contains $K = 3$ branches.

Inference. Each intervention branch receives the same initial observation and its own action sequence. Branches start from independent Gaussian noise.

We use 50 flow steps and apply shared-world guidance with a maximum strength of 0.01. The branches are therefore not coupled through shared random noise. Their interaction occurs only through the shared-world evidence gradient.

D.6 Sensitivity to Branch Count and Hyperparameters

Figure 4 reports sensitivity on Push Cube to branch count, shared-loss weight, maximum guidance strength, and the importance-sampling budget. Increasing the number of branches from $K = 2$ to 5 gradually increases Δ_{world} while causing only small changes in M-MSE. The tested settings therefore extend beyond the default three branches, but incur a larger shared-world gap as more branches are added.

Both λ_{shared} and $\lambda_{\max}$ exhibit a consistency–prediction trade-off: moderate values substantially reduce the gap, whereas stronger regularization or guidance gives diminishing consistency gains and begins to increase prediction error. Increasing the importance-sampling budget improves estimator stability: relative ESS increases and the variance of $\log \widehat{\mathcal{C}}_\tau$ decreases, with modest improvements beyond $M = 64$.

D.7 Computational Cost

We characterize inference cost on eight NVIDIA A100-SXM4-80GB GPUs. Each counterfactual group contains $K = 3$ branches, with 37 frames per branch at 240×240 resolution. All paths use FP32 inference and 50 flow steps.

Table 8: Inference cost on 8 NVIDIA A100-SXM4-80GB GPUs. Each group contains three branches with 37 frames at 240×240 resolution.

Compute path	Latency (s/group)	Peak mem. (GiB/GPU)	Relative	Throughput (groups/s)
ACWM-DiT	13.247 ± 0.012	2.403	$1.000\times$	0.603
OneWorld w/o guidance	13.245 ± 0.014	2.403	$1.000\times$	0.602
OneWorld full	$\mathbf{21.684 \pm 0.041}$	**7.864**	$\mathbf{1.637}\times$	**0.369**

Inference overhead. The no-guidance path has essentially the same inference cost as ACWM-DiT, because shared-world flow training changes the model parameters but does not add an additional sampling module. The full OneWorld path additionally evaluates the mechanism interpreter and the $M = 64$ shared-world evidence estimator at each guided flow step, resulting in higher latency and memory usage while retaining practical group-level throughput.

E Implementation of Shared-World Evaluation

This section describes the controlled environments and the simulator-based evaluator used for the main shared-world experiments.

The evaluator is independent of the learned physical mechanism interpreter. Generated videos are first converted into observable state trajectories. Their physical compatibility is then measured through environment replay and physical-parameter fitting.

E.1 Controlled Environment Construction

Push Cube and Push Rope. The Push Cube and Push Rope experiments use our reconstructed Pymunk environments following the corresponding interaction settings of ACWM-Phys. These environments are independent reconstructions and are not the original ACWM-Phys simulator implementation.

For both reconstructed environments, we use 128 training initial states, 32 validation initial states, and 32 test initial states. Each initial state produces one same-world group and one mismatched-world group, resulting in 256/64/64 groups for the train/validation/test splits.

Each group contains $K = 3$ intervention branches. Videos contain 37 frames rendered at 240×240 resolution and 10 Hz.

Before every branch is generated, the environment is reset to the same complete initial state. In a same-world group, the physical configuration is also reset to the same value before each action is applied.

For Push Cube, the physical configuration is controlled by two quantities: drag and elasticity. Each quantity takes three values spanning low, intermediate, and high regimes. Their Cartesian product produces nine physical configurations.

For Push Rope, the physical configuration is controlled by bending stiffness and bending damping. Each quantity likewise takes three values, producing nine physical configurations.

Pour Water. Pour Water is implemented using a separate particle-based fluid environment rather than the reconstructed Pymunk system.

We use 160/40/40 initial states for the train/validation/test splits, respectively. Each initial state contributes one same-world group and one mismatched-world group, giving 320/80/80 groups.

Each counterfactual group contains three action branches. Videos contain 37 frames rendered at 240×240 resolution and 10 Hz.

The hidden fluid configuration varies viscosity and particle cohesion. Each parameter takes four predefined values, yielding $4 \times 4 = 16$ candidate physical configurations.

Table 9: Controlled environments used for shared-world evaluation.

Environment	Initial states	Groups	K	Video	Physical parameters	Configs.
Push Cube	128/32/32	256/64/64	3	37 frames, 240^2, 10 Hz	drag, elasticity	3×3
Push Rope	128/32/32	256/64/64	3	37 frames, 240^2, 10 Hz	bending stiffness, bending damping	3×3
Pour Water	160/40/40	320/80/80	3	37 frames, 240^2, 10 Hz	viscosity, particle cohesion	4×4

E.2 Same-World and Mismatched-World Construction

For a same-world group, the simulator is reset to the same complete initial state and the same physical configuration before each intervention. Three different action sequences are then executed. Thus, the branches share the underlying physical world but may exhibit different trajectories because of their different actions.

For a mismatched-world group, the visible initial state is held fixed while branches are drawn from different physical configurations. The action sequences follow the same sampling procedure as those used in positive groups.

This construction is intended to prevent the mechanism classifier from relying on obvious differences in the initial scene. The distinction between positive and negative groups is instead determined by whether the action-conditioned outcomes are compatible with one common physical configuration.

No generated video from the evaluated world model is used to construct the group labels.

E.3 Frozen State Extractors

Shared-world metrics operate on observable state trajectories rather than directly on pixels.

A separate state extractor is trained for each environment using only simulator-rendered frames paired with ground-truth simulator states. The extractor is frozen before any generative model is evaluated, and the same frozen extractor is applied to all compared methods.

Push Cube. The observable state is

$$S_t = (x_t, y_t, \cos\theta_t, \sin\theta_t),$$

containing the two-dimensional cube center and orientation. A ResNet-18 (He et al., 2016) regressor predicts this state.

Push Rope. The rope is represented using 16 ordered keypoints sampled uniformly along its centerline. A heatmap-based keypoint network predicts their image-space positions.

Pour Water. The fluid state consists of a 16×16 occupancy map and the fluid centroid. We use a lightweight U-Net (Ronneberger et al., 2015) to estimate the occupancy representation and corresponding centroid.

Table 10: Validation accuracy of the frozen state extractors used by the simulator-based evaluator.

Environment	Observable state	Validation error
Push Cube	2D center and orientation	position RMSE 0.011, angle error 2.8°
Push Rope	16 ordered rope keypoints	mean point RMSE 0.017
Pour Water	16×16 occupancy and centroid	occupancy IoU 0.928, centroid RMSE 0.014

The state-extractor networks are never fine-tuned using outputs from ACWM-DiT, OneWorld, or any compared baseline.

E.4 Environment-Specific State Distances

The simulator-fitting objective uses an environment-specific state distance $d(\hat{S}_t, S_t)$.

Push Cube. Let p_t denote the cube center and θ_t its orientation. We use

$$d_{\text{cube}}(\hat{S}_t, S_t) = \frac{\|\hat{p}_t - p_t\|_2^2}{s_p^2} + 0.25\left[1 - \cos(\hat{\theta}_t - \theta_t)\right],$$

where s_p is the position normalization scale estimated from the training split.

The first term evaluates translation and the second term penalizes orientation disagreement while respecting angular periodicity.

Push Rope. For 16 ordered rope keypoints,

$$d_{\text{rope}}(\hat{S}_t, S_t) = \frac{1}{16}\sum_{j=1}^{16} \frac{\|\hat{p}_{t,j} - p_{t,j}\|_2^2}{s_r^2},$$

where s_r is fixed from the training split.

Pour Water. For the fluid environment, we combine occupancy mismatch and centroid displacement:

$$d_{\text{water}}(\hat{S}_t, S_t) = 0.6\left[1 - \text{IoU}(\hat{O}_t, O_t)\right] + 0.4\frac{\|\hat{c}_t - c_t\|_2^2}{s_c^2}.$$

Here O_t is the occupancy map, c_t is the fluid centroid, and s_c is the corresponding training-set normalization scale.

All normalization constants are fixed before test evaluation and are shared across all compared prediction methods.

E.5 Simulator Replay and Physical-Parameter Search

For each generated branch, the evaluator extracts an observable trajectory $\hat{S}^{(k)}$ and replays the corresponding action sequence in the environment under candidate physical configurations w.

The replay starts from the exact initial simulator state associated with the evaluated example. For each candidate w, the simulator produces

$$S^{(k)}(w),$$

and the branch fitting error is

$$\ell_k(w) = \frac{1}{T}\sum_{t=1}^{T} d\left(\hat{S}_t^{(k)}, S_t^{(k)}(w)\right).$$

For Push Cube, the candidate space is the Cartesian product of three drag values and three elasticity values, yielding nine candidate configurations.

For Push Rope, the search spans three bending-stiffness values and three bending-damping values, also yielding nine configurations.

For Pour Water, the evaluator searches the 4×4 grid of viscosity and cohesion configurations used by the particle-based environment.

For every evaluated method, the candidate grid, simulator replay, state extractor, and state-distance definition are identical.

E.6 Evaluator Grid-Density Sensitivity

The main evaluator uses the physical configurations defined by each environment. To test whether the reported shared-world gap is dominated by coarse parameter discretization, we repeat fitting on denser interpolation grids over the same physical ranges. The generated videos, state extractors, action sequences, and distance functions are fixed; only the evaluator grid is changed.

For Push Cube and Push Rope, we compare 3×3, 5×5, and 9×9 candidate grids. Intermediate grid points are obtained by uniform interpolation between the minimum and maximum values of each physical parameter. Table 11 reports the resulting Δ_{world} for ACWM-DiT and OneWorld.

Table 11: Sensitivity of the shared-world gap to evaluator grid density. Lower is better. The model outputs are fixed across all columns.

Environment	**Method**	3×3	5×5	9×9
Push Cube	ACWM-DiT	0.02253	0.02187	0.02161
	OneWorld	**0.00357**	**0.00341**	**0.00336**
Push Rope	ACWM-DiT	0.01389	0.01343	0.01326
	OneWorld	**0.00209**	**0.00197**	**0.00191**

Densifying the search grid slightly lowers the absolute gap for both methods, as expected when parameter fitting becomes more precise, but the change is small relative to the separation between ACWM-DiT and OneWorld. The same ordering is preserved at every tested density, indicating that the main consistency result is not an artifact of the default 3×3 evaluator grid.

E.7 Individual and Shared Physical Fits

The individual physical fitting error is

$$E_{\text{ind}} = \frac{1}{K}\sum_{k=1}^{K} \min_{w_k} \ell_k(w_k).$$

Each branch is therefore allowed to select its own best-fitting physical configuration. This measures whether each generated future is individually compatible with some physical explanation inside the evaluator family.

The shared physical fitting error is

$$E_{\text{shared}} = \min_{w} \frac{1}{K} \sum_{k=1}^{K} \ell_k(w).$$

Here all branches must use one common physical configuration.

The shared-world gap is

$$\Delta_{\text{world}} = E_{\text{shared}} - E_{\text{ind}}.$$

The gap measures the additional physical fitting cost induced by requiring one explanation to account for every intervention branch.

We report E_{ind}, E_{shared}, and Δ_{world} together so that cross-intervention consistency is interpreted alongside overall physical fit quality. The metric evaluates compatibility within the specified simulator family and candidate parameter space rather than uniqueness of the underlying physical mechanism.

E.8 Ground-Truth Physical Configuration Recovery

The shared-world gap measures whether several generated branches admit one common physical explanation, but consistency alone does not establish that the common explanation matches the physical configuration that generated the underlying world. Because the controlled simulators retain the ground-truth hidden configuration for every held-out same-world group, we additionally evaluate whether the jointly fitted configuration recovers that ground truth.

For held-out group i, let $w_i^\star$ denote the simulator configuration used to generate the underlying world. We first recover the shared physical configuration from the generated branches using the same simulator-fitting objective as in Eq. 12,

$$\hat{w}^{(i)}_{\text{shared}} = \arg\min_{w} \frac{1}{K} \sum_{k=1}^{K} \ell_{i,k}(w). \tag{13}$$

We then define exact physical-configuration recovery as

$$\text{RecoveryAcc} = \frac{1}{N} \sum_{i=1}^{N} \mathbb{I}\left[\hat{w}^{(i)}_{\text{shared}} = w_i^\star\right]. \tag{14}$$

If several candidate configurations have numerically identical minimum fitting error, the prediction is counted as correct when $w_i^\star$ is among the minimizing configurations. The candidate parameter grid, frozen state extractor, simulator replay, and state-distance function are identical to those used for the shared-world evaluation. Thus this metric introduces no additional learned evaluator and uses simulator ground truth only for scoring.

Table 12: Recovery accuracy (%) of ground-truth physical configurations on held-out same-world groups. Higher is better.

Method	**Push Cube ↑**	**Push Rope ↑**	**Pour Water ↑**	**Avg. ↑**
ACWM-DiT	59.4	62.5	50.0	57.3
Vid2World	62.5	65.6	52.5	60.2
CoCo	65.6	71.9	57.5	65.0
Twin Rollouts	65.6	68.8	55.0	63.1
Shared Noise	53.1	59.4	45.0	52.5
Posterior Alignment	<u>71.9</u>	<u>75.0</u>	<u>60.0</u>	<u>69.0</u>
OneWorld	**87.5**	**87.5**	**80.0**	**85.0**

Interpretation. Recovery accuracy complements Δ_{world} rather than replacing it. A low shared-world gap indicates that the intervention branches can be jointly explained by one physical configuration, whereas Table 12 asks whether that common configuration agrees with the simulator configuration that actually generated the world. OneWorld achieves the highest recovery accuracy in all three environments, with 87.5% on Push Cube, 87.5% on Push Rope, and 80.0% on Pour Water, for a macro-average of 85.0%. Relative to ACWM-DiT, this corresponds to gains of 28.1, 25.0, and 30.0 percentage points, respectively. The improvement is also substantial over Posterior Alignment, the strongest alternative in this evaluation, whose macro-average is 69.0%.

The environment-wise pattern is consistent with the difficulty of the underlying fitting problem. Push Rope gives the highest absolute recovery for most methods, while Pour Water remains more challenging because occupancy and centroid trajectories can be compatible with multiple nearby fluid configurations. Importantly, OneWorld retains a large margin on Pour Water, where its 80.0% recovery exceeds Posterior Alignment by 20.0 percentage points. This suggests that the reduction in shared-world gap is not explained only by forcing branches toward an arbitrary common configuration.

Taken together with Table 1, these results separate two properties that can otherwise be conflated. The shared-world gap measures whether the generated intervention branches agree on one physical explanation; physical-configuration recovery measures whether that explanation matches the known generating physics. The consistent improvement on both criteria supports the conclusion that OneWorld produces branches that are not only mutually compatible but also more faithful to the latent physical configuration of the controlled world. The reported average is the macro-average of the three environment-level recovery accuracies.

E.9 Generalization to Unseen Interpolated Physics

The main experiments evaluate physical configurations drawn from the discrete parameter grids used during training. We additionally evaluate whether the learned shared-world constraint generalizes to *intermediate* physical values that lie inside the training range but are never used during mechanism learning or world-model training.

To make the interpolation protocol comparable across environments, we report parameter values in range-normalized coordinates. For a physical parameter w_j with training-range endpoints $w_j^{\min}$ and $w_j^{\max}$, define

$$\tilde{w}_j = \frac{w_j - w_j^{\min}}{w_j^{\max} - w_j^{\min}}. \tag{15}$$

The normalized values therefore describe the relative location of the simulator-native parameter values inside the training range rather than replacing the native simulator configuration.

Interpolation split. For Push Cube and Push Rope, each physical factor has three training levels, which correspond to normalized coordinates $\{0, 0.5, 1\}$. Held-out interpolation uses the two midpoints $\{0.25, 0.75\}$. For Pour Water, each physical factor has four training levels, represented by $\{0, \frac{1}{3}, \frac{2}{3}, 1\}$, and interpolation uses the three adjacent midpoints $\{\frac{1}{6}, \frac{1}{2}, \frac{5}{6}\}$. The interpolation coordinates are excluded from all mechanism-learning and world-model training groups and are used only for held-out evaluation. Test initial states are also disjoint from the training and validation initial-state splits.

We use *joint interpolation*: both physical factors of a test world are drawn from the held-out intermediate set. Thus the Push Cube and Push Rope tests contain $2 \times 2 = 4$ unseen joint physical configurations, while Pour Water contains $3 \times 3 = 9$. This setting is stronger than varying a single unseen factor while keeping the second factor at a training value.

Table 13: Physics interpolation split. Values are shown in range-normalized parameter coordinates. Interpolation values are excluded from mechanism learning and world-model training and are used only for held-out evaluation.

Environment	Physical factor	Training values	Interpolation values
Push Cube	drag	$\{0,\ 0.5,\ 1\}$	$\{0.25,\ 0.75\}$
	elasticity	$\{0,\ 0.5,\ 1\}$	$\{0.25,\ 0.75\}$
Push Rope	bending stiffness	$\{0,\ 0.5,\ 1\}$	$\{0.25,\ 0.75\}$
	bending damping	$\{0,\ 0.5,\ 1\}$	$\{0.25,\ 0.75\}$
Pour Water	viscosity	$\{0,\ \frac{1}{3},\ \frac{2}{3},\ 1\}$	$\{\frac{1}{6},\ \frac{1}{2},\ \frac{5}{6}\}$
	particle cohesion	$\{0,\ \frac{1}{3},\ \frac{2}{3},\ 1\}$	$\{\frac{1}{6},\ \frac{1}{2},\ \frac{5}{6}\}$

Continuous-physics evaluator. For interpolation evaluation, the simulator-fitting candidate set is densified over the same physical ranges and explicitly includes every held-out interpolation value. The evaluator therefore does not force an unseen test world to snap to one of the discrete training configurations. We recover the shared physical configuration with Eq. 13 and measure both cross-branch consistency and physical-parameter accuracy.

Because exact recovery is unnecessarily brittle for continuous parameters, we report *normalized physical parameter error* (NPE). Let $\tilde{w}_i^\star$ be the range-normalized ground-truth configuration and $\tilde{\hat{w}}^{(i)}_{\text{shared}}$ the normalized jointly fitted configuration. We define

$$\text{NPE} = \frac{1}{N}\sum_{i=1}^{N}\left\|\tilde{\hat{w}}^{(i)}_{\text{shared}} - \tilde{w}_i^\star\right\|_2. \tag{16}$$

NPE is comparable across environments because each physical factor is normalized by its training range. Table 14 reports the interpolation Δ_{world} values separately for each environment and summarizes physical-parameter accuracy with Avg. NPE.

Table 14: Generalization to unseen interpolated physics. We report the shared-world gap and normalized physical parameter error (NPE) on held-out joint interpolation configurations. Lower is better for both metrics.

Method	Push Cube		Push Rope		Pour Water		Avg. NPE ↓
	Δ_{world} ↓	NPE ↓	Δ_{world} ↓	NPE ↓	Δ_{world} ↓	NPE ↓	
ACWM-DiT	0.02431	0.218	0.01542	0.194	0.03876	0.267	0.226
Vid2World	0.02046	0.193	0.01327	0.176	0.03418	0.239	0.203
CoCo	0.01683	0.164	0.01109	0.151	0.02874	0.208	0.174
Twin Rollouts	0.01812	0.178	0.01221	0.162	0.03136	0.221	0.187
Shared Noise	0.03268	0.241	0.01821	0.216	0.04357	0.291	0.249
Posterior Alignment	0.01407	0.137	0.00918	0.124	0.02384	0.181	0.147
OneWorld	**0.00462**	**0.079**	**0.00284**	**0.064**	**0.00713**	**0.103**	**0.082**

Interpretation. This experiment separates interpolation in the underlying physics from generalization to new initial scenes. Every evaluated test group uses both an initial state absent from training and physical parameter values that never appear during either mechanism learning or world-model training. The shared-world gap measures whether the generated branches remain compatible with one physical explanation at these unseen intermediate values, while NPE measures whether the common explanation is close to the actual continuously interpolated simulator configuration.

The two metrics should be interpreted jointly. A method can obtain a small shared-world gap by making its branches mutually compatible around an incorrect physical configuration; such a failure is exposed by high NPE. Conversely, low NPE together with a low shared-world gap indicates that

the model both preserves one common world across interventions and places that world near the correct intermediate physics. The environment-level Δ_{world} values show the consistency behavior in each dynamics regime, while Avg. NPE summarizes normalized physical-parameter accuracy across environments.

The joint-interpolation protocol also rules out a weaker explanation in which the model interpolates only one physical factor while relying on a second factor seen during training. Because both coordinates of every test configuration are held out, successful performance requires interpolation in the two-dimensional physical parameter space rather than recognition of one of the original discrete training configurations.

E.10 Independence of Training and Evaluation

The latent mechanism W used by OneWorld and the explicit simulator parameter vector w used by the evaluator are separate objects.

The learned interpreter is trained only to model compatibility among action–outcome branches. Its dimensions are not used by the simulator-based evaluator.

Conversely, the evaluator operates only on frozen state extractors, environment replay, and explicit candidate physical configurations. It does not access $q_\phi(W)$, $\pi_\phi(W \mid I_0)$, $\mathcal{C}_\tau$, or any intermediate feature from OneWorld.

Thus, an improvement in the reported shared-world gap cannot arise from directly evaluating predictions using the same learned compatibility function that supplied the training objective.