PHYS4D: INJECTING PHYSICAL STRUCTURE INTO VIDEO DIFFUSION VIA LIFTED RGB-D-MOTION 4D INTERFACES

Anonymous authors
Paper under double-blind review

ABSTRACT

Recent video diffusion models generate increasingly realistic videos, yet visual plausibility alone does not imply physical understanding. Physical laws govern evolving 3D scene states, while videos observe only their 2D projections. We introduce **Phys4D**, a structure-focused, appearance-constrained physics injection framework that transfers simulation-derived physical structure into pretrained video diffusion models while preserving their real-video generative prior. Phys4D operationalizes physics-aware video generation through a prior-preserving RGB-D-motion interface. Our key principle is **simulation teaches physics, not appearance**: simulation provides dense structural supervision, including depth, motion, and masks, while adaptation remains anchored to the pretrained real-video prior rather than drifting toward simulator-specific visual statistics. To this end, Phys4D exposes geometry and motion with lightweight prediction heads, injects local physical structure through gated geometry-motion supervision, and aligns sequence-level evolution using a simulation-grounded trajectory objective. Supported by a scalable coupled-physics simulation system, Phys4D improves external RGB physics scores and controlled geometry-motion diagnostics across multiple video generation backbones while maintaining visual quality. The simulated diagnostic split tests unseen templates within supported physical regimes. We plan to release code and datasets upon acceptance. Project page: `https://sensational-brioche-7657e7.netlify.app/`

1 INTRODUCTION

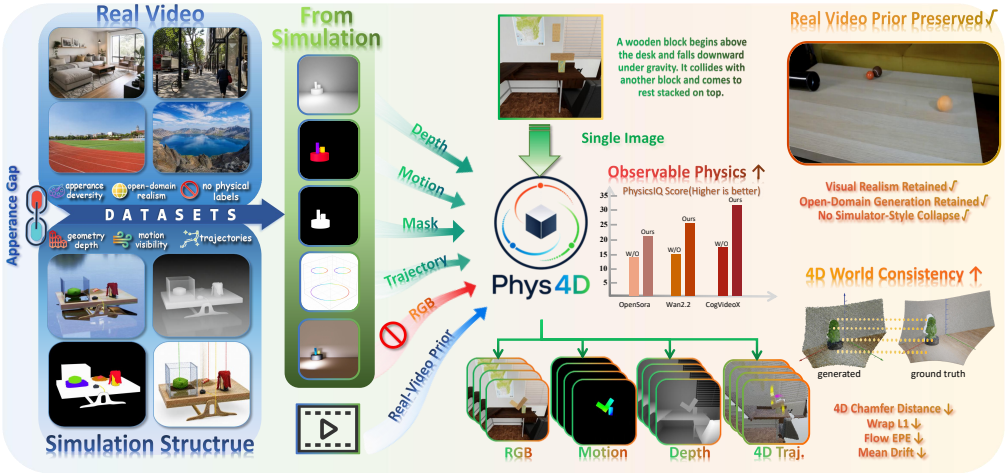

Figure 1: **Phys4D: simulation teaches physics, not appearance.** Phys4D uses simulation-derived depth, motion, masks, and trajectories to inject physical structure into pretrained video diffusion models through a 4D world interface, while blocking simulator RGB as the dominant appearance target. This improves observable physics and 4D geometry–motion consistency while preserving the real-video generative prior. The inset reports the Physics-IQ Original scores in Table 1.

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.

Recent video generation models (Liu et al., 2024b; Google DeepMind, 2024; Polyak et al., 2024; Chen et al., 2025c; Yu et al., 2024a;b; Huang et al., 2025b) have made remarkable progress in visual realism. Yet visual plausibility should not be mistaken for physical understanding (Zheng et al., 2025; Motamed et al., 2025). Physical laws govern world states in 3D, including geometry, depth ordering, object motion, contact, and deformation, whereas a video observes only their 2D projection. We therefore view physics-aware video generation as requiring **temporally consistent 3D scene evolution**. We operationalize this view through a scalable **RGB-D-motion interface**: throughout this paper, **"4D" refers to geometry and motion evolving over time, rather than full volumetric dynamic scene reconstruction.** This interface remains compatible with video diffusion models while exposing physical variables for geometry-level and motion-level supervision.

A central challenge is how to inject physical knowledge into video generation models. Fine-grained dynamics involve diverse physical regimes, including motion, contact, and multi-object coupling. Manually translating each regime into generator-specific losses, rewards, or constraints does not scale to open-ended interactions. We instead use physics-based simulation as an **executable form of human physical knowledge**: physical principles are encoded through solvers, constraints, contacts, materials, and coupling mechanisms, and rendered into video-compatible supervision. This provides dense, temporally aligned signals—including depth, motion, visibility, object masks, and trajectories.

However, simulation also introduces a key problem. Simulated videos are generated from procedural assets and renderer-specific statistics, making them less visually diverse than internet-scale real videos and mismatched in appearance. Directly fitting a video generator to such data risks transferring simulator-specific visual biases and degrading the pretrained real-video prior. Thus, simulation must be used selectively: **to prioritize physical structure while constraining appearance drift**.

This leads to **Phys4D**, a structure-focused, appearance-constrained physics injection framework that transfers simulation knowledge into pretrained video diffusion models while preserving their real-video generative prior. Phys4D proceeds in three stages. First, it constructs a **prior-preserving RGB-D-motion interface** by freezing the pretrained video backbone and training only lightweight depth and motion heads. These heads are supervised by pseudo-labels from strong monocular estimators on both self-generated videos and curated in-the-wild internet videos, allowing the model to expose geometry and motion without altering its RGB generator. Second, Phys4D injects local physical structure from simulation through **gated geometry-motion supervision**. Rather than directly fine-tuning the full generator on simulator videos, simulation-induced updates are confined to high-noise adapters and mask-weighted toward physics-relevant dynamic regions, where depth and motion supervision and GT-compensated warping couple geometry with motion. Third, Phys4D compares the completed generated evolution with a paired simulator rollout through a sequence-level geometric reward. It compares time-indexed dynamic-region point sets within a fixed one-frame temporal neighborhood and uses the terminal reward to update lightweight adapters.

To support these stages, we build a **scalable coupled-physics simulation system** that provides structural supervision for Phys4D. Unlike narrow single-regime simulation pipelines, our system asynchronously generates diverse interactions across rigid, deformable, fluid, and coupled dynamics, and exports unified geometry, motion, mask, visibility, and trajectory annotations. This design enables local geometry-motion injection and global trajectory-level alignment without relying on manually designed, category-specific physics rules.

We evaluate Phys4D from three complementary perspectives: external RGB video-level physics, simulation-reference geometry-motion diagnostics, and visual-prior preservation. The shared-extractor comparison isolates structure recoverable from generated RGB, while the native-interface readout includes specialized heads. A common RGB tracker recovers predicted worldlines from each generated video; these correspondence-aware diagnostics support improvements within the tested regimes, not out-of-distribution physics generalization or guaranteed object identity.

In summary, our contributions are as follows:

- We formulate physical knowledge injection for video generation as a **structure-transfer problem**, where simulation provides geometry, motion, and trajectory supervision while the pretrained real-video generative prior is preserved.

- We introduce **Phys4D**, a structure-focused physics injection framework that constructs a prior-preserving RGB-D-motion interface, injects local physical structure through gated geometry-motion supervision, and aligns global scene evolution through a unified trajectory-level policy objective.
- We build a **scalable coupled-physics simulator** and report gains on external RGB physics benchmarks and controlled geometry-motion diagnostics across multiple video backbones while maintaining visual quality.
- We provide an anonymized project page for paper resources and plan to release code and simulation assets upon acceptance.

2 Related Work

Physics-aware video generation and evaluation. Large-scale video generators achieve strong visual realism, but physical state remains largely implicit in RGB videos (Liu et al., 2024b; Polyak et al., 2024; Yang et al., 2025; Tencent Hunyuan Foundation Model Team, 2024; Wan et al., 2025). Recent benchmarks expose this gap through tests of physical commonsense, dynamics, and intrinsic faithfulness (Meng et al., 2025; Motamed et al., 2025; Bansal et al., 2024; Zheng et al., 2025). Physics-aware methods improve plausibility using motion priors, reinforcement learning, reasoning, or action-conditioned simulation interfaces (Xu et al., 2026; Zhang et al., 2026; Yuan et al., 2025; Liu et al., 2026). However, they typically operate on video-level outcomes, RGB-observable cues, or task-specific controls. Phys4D instead injects physical knowledge as structure: geometry, motion, and trajectory evolution.

4D scene evolution and simulation-based structural supervision. Dynamic scene modeling has developed rich 3D/4D representations, including neural fields, dynamic Gaussians, diffusion-based 4D reconstruction, and dynamic point maps (Pumarola et al., 2021; Park et al., 2021; Luiten et al., 2023; Wu et al., 2024a;b; Sucar et al., 2025). Physically grounded 4D generation further couples explicit geometry with simulation or physical priors (Xie et al., 2024; Huang et al., 2025a; Chen et al., 2025a; Li et al., 2025). These methods are effective for reconstruction, animation, and controlled 3D generation, but often rely on specialized representations or explicit 4D optimization. Phys4D instead uses simulation-derived structural supervision to train a prior-preserving RGB-D-motion interface for video diffusion.

3 Problem Definition

Let c denote the generation condition, which may be a text prompt p, an initial image I_{ref}, or an image-text pair (I_{ref}, p). Given c, our goal is to generate a video $\{I_t\}_{t=0}^{T}$ that is visually realistic while following physically plausible scene evolution. Since physical states such as geometry, depth ordering, contact, deformation, and motion are only partially observed through 2D videos, we formulate physics-aware video generation as a **structure-transfer problem**: simulation should supervise physical structure, rather than RGB appearance.

We expose physical structure through a compact **RGB-D-motion interface**. Let G_{θ_0} be a pretrained video diffusion model with a strong real-video prior. We seek an augmented model G_θ that preserves this prior while predicting depth, motion, and lifted 4D trajectories:

$$G_\theta(c, \epsilon) \longmapsto \Big(\{I_t\}_{t=0}^{T}, \{D_t\}_{t=0}^{T}, \{F_{t\to t+1}\}_{t=0}^{T-1}, \mathcal{T} \Big),$$

where $D_t \in \mathbb{R}^{H\times W}$ is depth, $F_{t\to t+1} \in \mathbb{R}^{H\times W\times 2}$ is inter-frame motion, and $\mathcal{T}$ denotes trajectories lifted from the predicted geometry and motion. Throughout this paper, "4D" refers to geometry and motion evolving over time, rather than full volumetric dynamic reconstruction.

Simulation provides structural supervision $\mathcal{Y}^{\mathrm{sim}}_{\mathrm{structure}} = \{D, F, M, \mathcal{T}\}^{\mathrm{sim}}$, covering depth, optical flow, masks, and trajectories. These signals encode executable physical knowledge, while simulated RGB frames are used only through constrained adaptation rather than as the generator's dominant fitting target. Thus, Phys4D performs **structure-focused, appearance-constrained physical knowledge injection**: it preserves the pretrained model's open-domain visual prior while improving depth, motion, and trajectory consistency. Detailed definitions of the lifted geometry-motion quantities are provided in Appendix B.

4 SCALABLE PHYSICS SIMULATION FOR 4D SUPERVISION

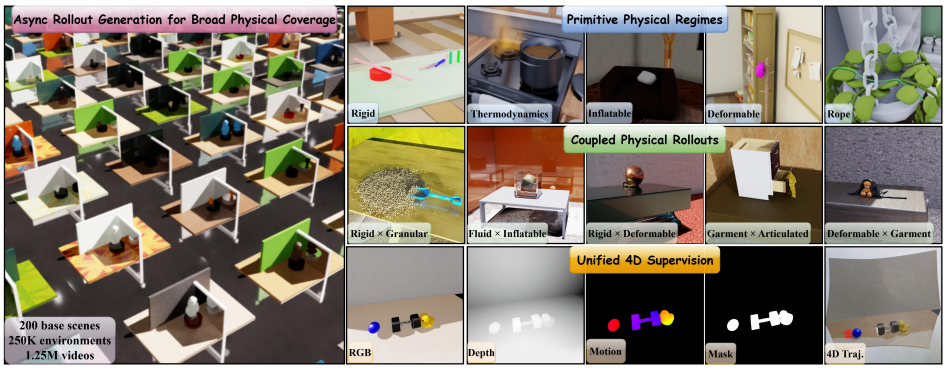

Figure 2: **Simulation as a scalable physical teacher. Left:** Asynchronous rollouts expand 200 scenes into 250K environments and 1.25M videos. **Top right:** The simulator covers primitive physical regimes. **Middle right:** These regimes are composed into coupled physical rollouts. **Bottom right:** Each rollout exports RGB, depth, motion, masks, and 4D point trajectories.

Real videos provide rich appearance priors but rarely expose dense 4D physical states such as depth, motion, and object masks. We therefore use simulation not as a target visual domain, but as an executable physical teacher that converts solvers and constraints into video-compatible structural supervision. To scale this supervision, we employ **asynchronous rollout generation**, where independent scenes run in parallel and resources are recycled upon completion. This covers randomized physical parameters, object configurations, and coupled interactions, producing RGB-D-motion-trajectory annotations for Stage-II local geometry-motion injection and Stage-III sequence-level alignment with local temporal matching while preserving the pretrained real-video prior.

Unified coupled-physics framework. Based on Isaac Sim (NVIDIA Corporation, 2024) and GarmentLab (Lu et al., 2024b;a), we integrate rigid-body dynamics, PBD, FEM, and task-specific solvers into a unified scene-level framework. It supports isolated regimes such as rigid objects, garments, fluids, deformables, ropes, granular materials, and thermodynamic state changes, as well as cross-regime coupling including rigid-fluid, rigid-garment, rigid-deformable, articulated-soft, and rigid-granular interactions. These coupled rollouts provide geometry, motion, visibility, and trajectory supervision for Phys4D. Detailed simulation settings are provided in Appendix K.

Domain randomization and 4D supervision. We randomize physical parameters, interaction layouts, geometry, and visual factors to learn invariant geometry-motion regularities rather than renderer-specific shortcuts. Randomizing mass, friction, restitution, stiffness, gravity, and scale exposes regimes governed by shared physical laws, while varying layouts and coupled interactions prevents memorization of fixed motion templates. All rollouts are exported through a unified 4D supervision interface with RGB frames, depth, scene flow, dynamic masks, camera poses, and spatiotemporal point trajectories, abstracting solver-specific states into common geometry-motion-trajectory supervision. With asynchronous generation, 200 base scenes are expanded into approximately 250K environments; five 10-second camera rollouts per environment yield 1.25M annotated videos, about 3.5K hours, and over 15TB of multi-modal annotations.

5 METHOD

Phys4D injects simulation-derived physical structure into a pretrained video diffusion model through a prior-preserving RGB-D-motion interface (Fig. 3). It proceeds in three stages: Stage I (Sec. 5.1) freezes the video backbone and trains lightweight depth and motion heads; Stage II (Sec. 5.2) injects local geometry-motion structure through high-noise, mask-gated adapters; and Stage III (Sec. 5.3) aligns generated 4D trajectories with simulator trajectories using a trajectory-level reward.

5.1 STAGE I: PRIOR-PRESERVING 4D INTERFACE CONSTRUCTION

Before physics injection, Stage I exposes geometry and motion through a prior-preserving RGB-D-motion interface. It uses no simulation data: the video backbone is frozen, and only lightweight depth and motion heads are trained.

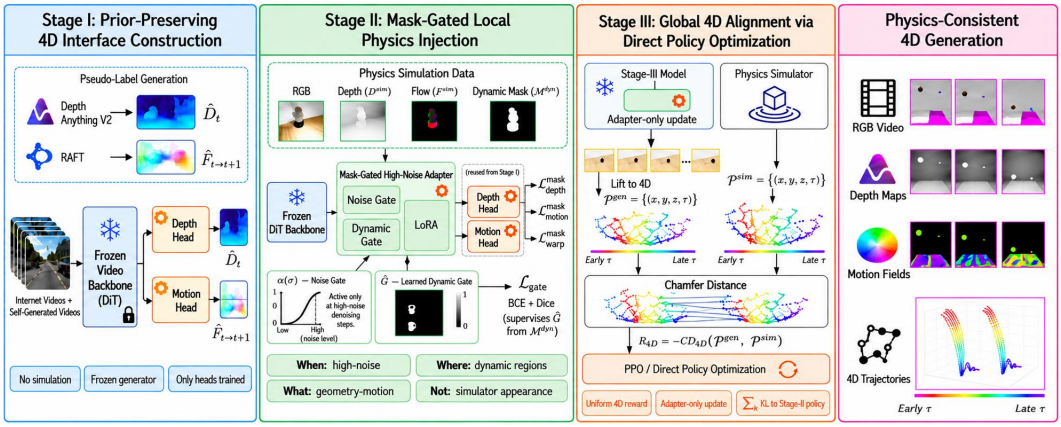

Figure 3: **Overview of the Phys4D training pipeline. Stage I** freezes the video DiT and trains depth/motion heads to build an RGB-D-motion interface. **Stage II** injects local physical structure through mask-gated high-noise adapters with simulation depth, motion, and GT-compensated warp supervision. **Stage III** aligns completed videos with paired simulation using a locally time-relaxed geometric reward.

Model Architecture. We attach two auxiliary heads to a pretrained DiT-style video diffusion backbone. The motion head predicts inter-frame optical flow $\hat{F}_{t\to t+1}$, and the depth head predicts per-frame depth $\hat{D}_t$. Following Chen et al. (2025b), the depth head incorporates temporal attention and multi-scale feature fusion for temporally coherent depth estimation. The DiT backbone remains frozen throughout Stage I; only the auxiliary heads are optimized. Details are provided in Appendix C.1.

Data Sources. We train the heads with pseudo labels on curated internet videos and self-generated videos from the pretrained model. Internet videos provide real-world diversity, while self-generated videos align the heads with the model's output distribution. Pseudo depth and flow are obtained from off-the-shelf estimators (Chen et al., 2025b; Teed & Deng, 2020), with confidence masks and per-clip depth normalization for denoising; details are in Appendix A.2.

Training Objective. Stage I uses supervised regression to match predicted depth and flow to the pseudo labels. The base video generation objective is unchanged, and gradients are applied only to the auxiliary heads. For details please refer to Appendix C.2.

5.2 Stage II: Mask-Gated Local Physics Injection

Stage I exposes an RGB-D-motion interface, but it does not teach physical behavior. The key idea of Stage II is to inject simulation-derived physical structure through a controlled pathway, rather than fine-tuning the generator to imitate simulator appearance. We restrict simulation supervision along three axes: **when to learn**, **where to learn**, and **what to learn**. Specifically, simulation-induced updates are applied only through lightweight physics adapters, activated at high-noise structure-forming denoising steps, and concentrated on dynamic regions where physical changes occur. The pretrained video backbone remains **frozen**; gradients from simulation-derived losses update only the **physics adapters**, **learned gates**, and **auxiliary geometry-motion heads**.

Where and when to inject: mask-gated physics adapter. We introduce a LoRA-style residual adapter as the **only trainable pathway** through which simulation modifies the generator. Given frozen backbone features h_ℓ at layer ℓ, the adapted features are

$$h'_\ell = h_\ell + \alpha(\sigma)\,\hat{G}_\ell \odot A_\ell(h_\ell), \tag{1}$$

where A_ℓ is a lightweight physics adapter, σ is the denoising noise level, $\alpha(\sigma)$ is a noise gate, and $\hat{G}_\ell$ is a learned dynamic-region gate. The noise gate controls **when** simulation can affect generation: physics supervision is injected at high-noise steps that determine coarse structure, layout, and motion, while low-noise appearance-refinement steps remain dominated by the pretrained real-video prior. The dynamic gate controls **where** the adapter acts: simulator object and dynamic masks supervise $\hat{G}_\ell$ during training, but no simulator masks are required at inference. Thus, simulation-induced adaptation is localized in **denoising time**, **image space**, and **parameter space**.

What to learn: geometry, motion, and temporal consistency. Simulation provides depth, optical flow or projected motion, visibility, and dynamic-region annotations. We use these signals to supervise **physical structure while constraining appearance drift**. Dynamic masks are used as soft weights, so the model preserves global scene geometry while emphasizing object, deformation, and interaction regions. The depth head is supervised by simulator depth, and the motion head is supervised by simulator flow or projected motion, with visibility masks removing occluded or invalid correspondences. The same dynamic annotations also train the gate to identify physics-relevant regions from model features.

Per-frame depth and motion supervision alone do not guarantee coherent depth evolution. We therefore use simulator flow to establish fixed correspondences and match signed predicted and simulator warp residuals. Let $r_t^{\text{pred}}(u) = \mathcal{W}(\hat{D}_t, F_{t\to t+1}^{\text{sim}})(u) - \hat{D}_{t+1}(u)$ and $r_t^{\text{sim}}(u) = \mathcal{W}(D_t^{\text{sim}}, F_{t\to t+1}^{\text{sim}})(u) - D_{t+1}^{\text{sim}}(u)$. Our **GT-compensated geometry warp** is

$$\mathcal{L}_{geom_warp}^{mask} = \frac{\sum_{t,u} m_t(u) w_t(u) \left| r_t^{\text{pred}}(u) - r_t^{\text{sim}}(u) \right|}{\sum_{t,u} m_t(u) w_t(u) + \epsilon}, \tag{2}$$

where m_t excludes invalid and occluded correspondences and w_t emphasizes dynamic regions, both on the warp-output grid. Both residuals use the same sampling operator. Unlike scalar-depth invariance, this objective permits real depth changes from object or camera motion; equal predicted and simulator depths give zero residual difference. Per-frame metric depth supervision anchors absolute depth. The target follows the stable-error motivation of Chen et al. (2025b), while retaining the sign of each residual.

We also include a weak **RGB warp self-consistency** term on generated frames to stabilize temporal appearance. Importantly, this term does not use simulator-rendered RGB as a photometric target; it only encourages adjacent generated frames to be consistent under motion. Thus, Stage II learns local depth, motion, and temporal geometry-motion consistency while avoiding direct fitting to simulator RGB appearance.

Overall objective. The Stage-II objective combines high-noise adapter training, mask-weighted geometry-motion supervision, generated-frame RGB self-consistency, and gate learning:

$$\mathcal{L}_{S2} = \mathcal{L}_{FM}^{high} + \lambda_R \mathcal{L}_{rgb_warp}^{self} + \lambda_D \mathcal{L}_{depth}^{mask} + \lambda_F \mathcal{L}_{motion}^{mask} + \lambda_W \mathcal{L}_{geom_warp}^{mask} + \lambda_G \mathcal{L}_{gate}. \tag{3}$$

Here the high-noise flow-matching term $\mathcal{L}_{FM}^{high}$ fits latents encoded from simulator-rendered videos through the adapter pathway. Table 6 distinguishes this RGB target from structural annotations and generated-frame RGB self-consistency. All gradients from simulation-derived losses are blocked from the frozen backbone and update only the lightweight adapters, learned gates, and auxiliary heads. Stage II therefore teaches **local geometry-motion physics** through a controlled structural pathway. Details are provided in Appendix C.3–C.5.

5.3 Stage III: Simulation-Grounded 4D Alignment by Direct Policy Optimization

Stage II injects local geometry-motion consistency, but adjacent-frame losses do not fully determine whether the completed generated evolution follows the intended physical trajectory. Stage III therefore performs an offline simulation-grounded trajectory alignment stage. Importantly, Stage III is performed only on simulation-paired training scenes, where the simulator provides the corresponding ground-truth 4D trajectory under the same initial state, camera, physical parameters, and prompt. It is not an online reward applied to arbitrary real images or open-domain prompts.

Simulation-Paired Rollouts. For each Stage-III rollout, we sample a simulation scene with a known scene ID, initial state, asset configuration, physical parameters, camera intrinsics and extrinsics, and text prompt. The generated video is conditioned on the simulator-rendered initial frame and the same text description. The corresponding reference trajectory P_{sim} is the simulator trajectory from this same rollout. Thus, P_{sim} is available only because the rollout is paired with a simulator state, not because arbitrary generated videos have ground-truth physical trajectories.

Given a generated video, we use the frozen depth and motion heads from the Stage-II checkpoint to construct valid dynamic-region 3D point sets P_i; the paired simulator gives Q_j. Both use the same reference coordinates and spatial scale. Write $C_{ij} = \text{CD}_{3D}(P_i, Q_j)$ for the symmetric 3D Chamfer

cost. For video-frame timestamps τ_i and fixed radius $r = 1$, we define

$$c_{\lambda_t}(i,j) = \begin{cases} C_{ij} + ((\tau_i - \tau_j)/\lambda_t)^2, & \lambda_t > 0,\ |i-j| \le 1, \\ C_{ii}, & \lambda_t = 0,\ i = j, \\ +\infty, & \text{otherwise.} \end{cases} \tag{4}$$

Thus λ_t changes the cost of an allowed one-frame offset, never the candidate window. With N video frames, the sequence discrepancy and bounded terminal reward are

$$\mathcal{D}_{\text{align}}^{(1,\lambda_t)}(P,Q) = \frac{1}{2N}\left[\sum_{i=0}^{N-1} \min_j c_{\lambda_t}(i,j) + \sum_{j=0}^{N-1} \min_i c_{\lambda_t}(i,j)\right], \tag{5}$$

$$R_{4D}(V;s) = 1 - 2\,\text{clip}\left(\frac{\mathcal{D}_{\text{align}}^{(1,\lambda_t)}(P,Q)}{d_{\text{cap}}}, 0, 1\right), \quad d_{\text{cap}} > 0. \tag{6}$$

At $\lambda_t = 0$, this discrepancy is exactly the mean same-frame 3D Chamfer cost. The scale d_{cap} is calibrated on training-side rollouts and then fixed. Local matches are selected independently, so the reward does not impose globally monotone correspondences or persistent point identities. Details are given in Appendix C.6.

Direct Policy Optimization. We formulate the stochastic reverse denoising process as a policy $\pi_{\theta+\phi}$, where θ denotes the frozen pretrained video backbone and ϕ denotes the lightweight Stage-II LoRA adapters. Stage III updates only these adapters:

$$\max_{\phi}\ \mathbb{E}_{s\sim\mathcal{S}_{sim},\, V\sim\pi_{\theta+\phi}(\cdot|I_0^{sim},c_s)}\left[R_{4D}(V;s) - \beta D_{\text{KL}}\left(\pi_{\theta+\phi}\|\pi_{\theta+\phi_{S2}}\right)\right], \tag{7}$$

where ϕ_{S2} is the Stage-II checkpoint, I_0^{sim} is the simulator-rendered initial frame, and c_s is the paired text description.

To avoid reward hacking, the depth head, motion head, and learned gate module are frozen during Stage III. Only the Stage-II LoRA adapters are optimized. After Stage-III training, the learned adapters are used for ordinary inference. No simulator trajectory, simulator depth, simulator mask, or P_{sim} is required at inference time. In this sense, Stage III learns adapter parameters from simulation-paired trajectory alignment and transfers the learned trajectory prior to open-domain generation.

6 Experiments

Our central hypothesis is that simulation can transfer physical structure into a pretrained video generator without replacing its real-video generative prior. In this section, we aim to answer the following research questions through our experiments:

RQ1: Does Phys4D improve observable physical plausibility in generated videos?
RQ2: Does Phys4D improve 4D world-level geometry-motion-trajectory consistency?
RQ3: Does Phys4D preserve the pretrained real-video generative prior?
RQ4: Do physics gains come from physical-structure supervision rather than simulator RGB fitting?

6.1 Simulation-Derived Physics Transfers to Generated Videos

Benchmarks and protocols. We evaluate observable video-level physical plausibility on three external diagnostics: Physics-IQ (Motamed et al., 2025), VBench-2.0 Physics (Zheng et al., 2025), and PhyGenBench (Meng et al., 2025). Following Sec. 3, we denote the generation condition as c. Physics-IQ follows a switch-frame I2V protocol, where $c = I_{\text{ref}}$ or $c = (I_{\text{ref}}, p)$ when a prompt is available. VBench-2.0 Physics and PhyGenBench follow prompt-based T2V protocols, where $c = p$. Detailed experiment settings are provided in Appendix F.2.

Implementation. We apply Phys4D to CogVideoX-5B (Yang et al., 2025), WAN2.2-5B (Wan et al., 2025), and Open-Sora V1.2 (Zheng et al., 2024). For Physics-IQ, we generate 5-second continuations from the provided switch frame. For VBench-2.0 Physics and PhyGenBench, we use their official prompt suites and evaluation pipelines.

Table 1: **RQ1: External video-level physics evaluation.** Phys4D improves observable physical plausibility across three backbones and three independent physics diagnostics. Physics-IQ is reported under our matched switch-frame I2V protocol.

Backbone	Params	Method	Physics-IQ MSE ↓	Physics-IQ Score ↑	VBench2.0-Physics ↑	Phygenbench ↑
CogVideoX	5B	Base	0.013 ± 0.002	18.8 ± 1.4	0.4849 ± 0.014	0.45 ± 0.02
CogVideoX	5B	+ Phys4D	$\mathbf{0.008 \pm 0.001}$	$\mathbf{32.8 \pm 1.9}$	$\mathbf{0.5954 \pm 0.018}$	$\mathbf{0.58 \pm 0.03}$
WAN2.2	5B	Base	0.016 ± 0.002	16.8 ± 1.3	0.4928 ± 0.015	0.45 ± 0.02
WAN2.2	5B	+ Phys4D	$\mathbf{0.013 \pm 0.001}$	$\mathbf{30.9 \pm 1.8}$	$\mathbf{0.6247 \pm 0.017}$	$\mathbf{0.57 \pm 0.03}$
Open-Sora V1.2	1.1B	Base	0.021 ± 0.003	14.5 ± 1.2	0.4623 ± 0.016	0.44 ± 0.02
Open-Sora V1.2	1.1B	+ Phys4D	$\mathbf{0.011 \pm 0.002}$	$\mathbf{24.7 \pm 1.6}$	$\mathbf{0.5899 \pm 0.019}$	$\mathbf{0.57 \pm 0.03}$

Results. Table 1 shows consistent gains from Phys4D across three backbones and three external physics diagnostics. This suggests that simulation-derived physical structure transfers to generated videos beyond simulator-specific metrics. Additional analysis is provided in Appendix F.1. Qualitative examples are in Appendix E (Fig. 6).

6.2 RQ2: Phys4D Improves 4D Geometry–Motion–Trajectory Consistency

Diagnostics. We test separately the structure recoverable from generated RGB and the quality of Phys4D's native RGB-D-motion interface. Simulated scenes come from 20 templates held out before rollout generation; assets, solvers, and parameter ranges may overlap with training. This tests unseen templates within supported regimes, not out-of-distribution physics. The diagnostics cover per-frame geometry, local depth/flow consistency, set-level 4D alignment, worldlines, and novel-time continuity (Appendix H).

Implementation. The shared-extractor comparison applies the same external depth and dense-flow estimators, DAV2 (Yang et al., 2024a) and SEA-RAFT (Wang et al., 2024), to WAN2.2 Base and Phys4D-generated RGB. The native-interface row uses Phys4D depth/flow outputs. For worldlines, the evaluation applies the same frozen CoTracker3 (Karaev et al., 2024) to each generated video from common first-frame queries, then lifts its tracks with the depth source of each row. The two Phys4D readouts share the same RGB video and 2D tracks; simulator trajectories are references, never predicted-track correspondences. The reported worldline scores use this tracker protocol (Appendix H.4).

Table 2: **RQ2: Shared-extractor RGB and native-interface diagnostics.** OTF uses the same external DAV2/SEA-RAFT readout on generated RGB. The native row uses Phys4D heads. The worldline protocol uses a shared RGB tracker and row-specific depth; Mean Drift and Fail Rate are measured from RGB-tracked trajectories against simulator references. Simulator annotations supply references; full definitions are in Appendix H.

Backbone	Method	AbsRel ↓	Warp L1 ↓	Flow EPE ↓	4D Chamfer ↓	Mean Drift ↓	Fail Rate ↓	Novel Depth ↓	Novel Warp ↓
WAN2.2	Base + OTF	0.3929	0.7990	1.2516	0.5058	0.5369	12.38%	0.5841	1.1076
CogVideoX	Base + OTF	0.3483	0.7054	1.2343	0.4923	0.5239	11.34%	0.5432	1.1254
Open-Sora V1.2	Base + OTF	0.4216	0.7436	1.2718	0.5286	0.5637	13.54%	0.6534	1.1573
Phys4D + WAN2.2	RGB+OTF	0.3347	0.6698	1.0824	0.4873	0.5126	10.91%	0.5527	1.0954
Phys4D + WAN2.2	Native interface	**0.2386**	**0.4558**	**0.4478**	**0.4159**	**0.4312**	**8.36%**	**0.4746**	**1.0289**

Results. With the external readout fixed, the matched WAN2.2 RGB comparison changes AbsRel from 0.3929 to 0.3347, Flow EPE from 1.2516 to 1.0824, and 4D Chamfer from 0.5058 to 0.4873. The native-interface row further improves these structural measurements, but that additional gap is not solely a change in generated RGB. With the same external depth readout, RGB-tracked Mean Drift and Fail Rate improve from 0.5369 and 12.38% to 0.5126 and 10.91%; native depth on the same Phys4D tracks gives 0.4312 and 8.36%. These are correspondence-aware diagnostics, not a guarantee of persistent object identity. External RGB physics evidence is reported in RQ1; further diagnostics and a qualitative point-cloud visualization appear in Appendix H (Fig. 8).

6.3 RQ3: Phys4D Preserves the Real-Video Visual Prior

Evaluation and implementation. We evaluate whether simulation-derived physics injection harms the pretrained real-video visual prior using VBench (Huang et al., 2023). For each backbone, we compare the base model and its Phys4D-enhanced counterpart under the same prompts, sampling

settings, and evaluation pipeline. We report *Aesthetic Quality*, *Imaging Quality*, *Color*, *Appearance Style*, and *Human Action*, and define **Visual Avg.** as the average of the first three metrics.

Table 3: **RQ3: VBench visual prior preservation.** We evaluate whether Phys4D preserves the pretrained real-video visual prior using VBench 1.0. **Visual Avg.** averages Aesthetic Quality, Imaging Quality, and Color. Phys4D keeps visual quality close to the base model while maintaining or improving Appearance Style and Human Action.

Backbone	Params	Method	Aesthetic Quality ↑	Imaging Quality ↑	Color ↑	Visual Avg. ↑	Appearance Style ↑	Human Action ↑
CogVideoX	5B	Base	0.6124	0.6931	0.9018	0.7358	0.2107	0.7452
CogVideoX	5B	+ Phys4D	0.6097	0.6886	0.8995	0.7326	**0.2184**	**0.7728**
Open-Sora V1.2	1.1B	Base	0.5948	0.6715	0.8842	0.7168	0.1972	0.6872
Open-Sora V1.2	1.1B	+ Phys4D	0.5917	0.6782	0.8821	0.7173	**0.2011**	**0.7023**
WAN2.2	5B	Base	0.6286	0.7164	0.9112	0.7521	0.2139	0.7433
WAN2.2	5B	+ Phys4D	**0.6351**	0.7119	0.9087	0.7519	**0.2162**	**0.7712**

Results. Table 3 shows that Phys4D keeps Visual Avg. nearly unchanged across all three backbones, while consistently maintaining or improving Appearance Style and Human Action. This indicates that Phys4D preserves open-domain visual quality. For more details, see Appendix F.8.

6.4 RQ4: Physical Structure Transfer Without Simulator Appearance Fitting

Setup. We ask whether the observed physical improvements require fitting simulator-video RGB targets. Table 4 compares WAN2.2-5B variants on Physics-IQ Original, which scores generated RGB continuations of real-world interactions without calling our depth/motion heads or simulator annotations. Here RGB-FM means flow matching against simulator-rendered video latents, not initial-image conditioning or generated-frame RGB self-consistency. The attribution comparison concerns Stage II; the Stage-II+III row gives full-method context. Appendix G provides supporting stage, loss, head, and temporal analyses and control configurations.

Results. Without simulator-video RGB-FM, structural training scores 25.7, retaining $8.9/11.0 \approx 81\%$ of the Stage-II gain over the 16.8 base; full Stage II scores 27.8. A headless high-noise RGB-FM variant scores 21.4. Direct simulator-RGB LoRA scores 24.2 but lowers VB2 Prior Avg. from 0.586 to 0.528; full Phys4D scores 30.9 with Prior Avg. 0.581. These interacting controls support structural supervision as important, not an isolated causal estimate of simulator-appearance effects.

Table 4: **RQ4: Physical structure Without simulator RGB adaptation.** WAN2.2-5B, Physics-IQ Original. The first five rows exclude Stage III; the last gives full-method context. Δ is relative to the base score. VB2 Prior Avg. uses non-physics VBench-2.0 dimensions; Control definitions are in Appendix G.

Training variant	Simulator-video RGB-FM target	Depth/motion branches	Physics-IQ ↑	Δ	VB2 Prior Avg. ↑
Base WAN2.2-5B	×	×	16.8	–	0.586
Direct simulator-RGB LoRA	✓ (all noise)	×	24.2	+7.4	0.528
Stage II w/o depth & flow	✓ (high noise)	×	21.4	+4.6	0.543
Stage II w/o RGB-FM	×	✓	**25.7**	**+8.9**	0.583
Full Stage II	✓ (high noise)	✓	**27.8**	**+11.0**	0.581
Full Phys4D (Stage II+III)	✓ (high noise)*	✓	30.9	+14.1	0.581

7 Conclusion

We presented Phys4D, which uses simulation as a teacher of physical structure rather than a target visual domain. Through an RGB-D-motion interface, it transfers geometry, motion, and temporal structure while preserving pretrained visual priors. Experiments across multiple backbones show improved physical plausibility with largely unchanged visual quality. Limitations and broader impacts are discussed in Appendix L.

AI USE STATEMENT

Generative AI tools assisted with manuscript language editing, LaTeX preparation, and consistency checks. The authors produced the experimental measurements and take responsibility for the mathematical definitions, reported results, and all AI-assisted content.

ETHICS STATEMENT

Stage I trains on curated internet videos and on videos self-generated by the pretrained models; for internet videos, we retain only clips with permissive licenses when license metadata is available, exclude private or personal content, and provide a removal mechanism for released metadata (Appendix A.2). All physics supervision in Stages II and III comes from simulation. As with any improvement in video generation quality, Phys4D could be misused to produce deceptive or misleading synthetic media. We discuss this risk, our responsible-release plan, and the limitations of this work in Appendix L.

REPRODUCIBILITY STATEMENT

The model architecture and the objectives of all three stages are specified in Sec. 5 and Appendix C, with the Stage-III policy-optimization formulation in Appendix I. Training protocols, compute cost, and inference overhead are reported in Appendix D. Data sources and filtering are described in Appendix A, and simulation solvers and parameter ranges in Appendix K (Table 25). Evaluation protocols for all benchmarks, ablations, and 4D diagnostics are given in Appendices F, G, and H. Paper resources are hosted on the anonymized project page linked in the abstract, and the release plan and licensing are described in the Appendix Overview.

REFERENCES

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. *arXiv preprint arXiv:2406.03520*, 2024.

Hritik Bansal et al. Videophy-2: A challenging action-centric physical commonsense evaluation in video generation. *arXiv preprint arXiv:2503.06800*, 2025.

Jan Bender, Matthias Müller, and Miles Macklin. Position-based simulation methods in computer graphics. In *Eurographics*, 2015.

Yoshua Bengio, Jérôme Louradour, Ronan Collobert, and Jason Weston. Curriculum learning. In *Proceedings of the 26th Annual International Conference on Machine Learning*, pp. 41–48, 2009.

Bharat Lal Bhatnagar, Xianghui Xie, Ilya A. Petrov, Cristian Sminchisescu, Christian Theobalt, and Gerard Pons-Moll. Behave: Dataset and method for tracking human object interactions. *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2022.

Kevin Black et al. Training diffusion models with reinforcement learning. *arXiv preprint arXiv:2305.13301*, 2023.

Andreas Blattmann et al. Align your latents: High-resolution video synthesis with latent diffusion models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2023.

Boyuan Chen, Hanxiao Jiang, Shaowei Liu, Saurabh Gupta, Yunzhu Li, Hao Zhao, and Shenlong Wang. Physgen3d: Crafting a miniature interactive world from a single image. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2025a.

Sili Chen, Hengkai Guo, Shengnan Zhu, Feihu Zhang, Zilong Huang, Jiashi Feng, and Bingyi Kang. Video depth anything: Consistent depth estimation for super-long videos. *2025 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 22831–22840, 2025b.

Yabo Chen, Yuanzhi Liang, Jiepeng Wang, Tingxi Chen, Junfei Cheng, Zixiao Gu, Yuyang Huang, Zicheng Jiang, Wei Li, Tian Li, et al. Teleworld: Towards dynamic multimodal synthesis with a 4d world model. *arXiv preprint arXiv:2601.00051*, 2025c.

Philippe G. Ciarlet. The finite element method for elliptic problems. In *Classics in applied mathematics*, 2002.

Carl Doersch et al. Tapir: Tracking any point with per-frame initialization and temporal refinement. *arXiv preprint arXiv:2306.08637*, 2023.

Google DeepMind. State-of-the-art video and image generation with veo 2 and imagen 3. `https://blog.google/technology/google-labs/video-image-generation-update-december-2024/`, 2024.

Kristen Grauman, Andrew Westbury, Lorenzo Torresani, Kris Kitani, Jitendra Malik, Triantafyllos Afouras, Kumar Ashutosh, et al. Ego-exo4d: Understanding skilled human activity from first- and third-person perspectives. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2024.

Klaus Greff et al. Kubric: A scalable dataset generator. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2022.

Jonathan Ho et al. Video diffusion models. In *Advances in Neural Information Processing Systems*, 2022.

Edward J. Hu et al. Lora: Low-rank adaptation of large language models. *arXiv preprint arXiv:2106.09685*, 2021.

Yuanming Hu et al. A moving least squares material point method with displacement discontinuity and two-way rigid body coupling. *ACM Transactions on Graphics*, 37(4):150:1–150:14, 2018.

Tianyu Huang, Haoze Zhang, Yihan Zeng, Zhilu Zhang, Hui Li, Wangmeng Zuo, and Rynson WH Lau. Dreamphysics: Learning physical properties of dynamic 3d gaussians with video diffusion priors. *Proceedings of the AAAI Conference on Artificial Intelligence*, 2025a.

Tianyu Huang, Wangguandong Zheng, Tengfei Wang, Yuhao Liu, Zhenwei Wang, Junta Wu, Jie Jiang, Hui Li, Rynson W. H. Lau, Wangmeng Zuo, and Chunchao Guo. Voyager: Long-range and world-consistent video diffusion for explorable 3d scene generation. *ACM Transactions on Graphics (TOG)*, 44:1 – 15, 2025b.

Yinghao Huang, Omid Taheri, Michael J. Black, and Dimitrios Tzionas. Intercap: Joint markerless 3d tracking of humans and objects in interaction. In *German Conference on Pattern Recognition (GCPR)*, 2022.

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, 2023. URL https://arxiv.org/abs/2311.17982.

Ziqi Huang et al. Vbench++: Comprehensive and versatile benchmark suite for video generative models. *arXiv preprint arXiv:2411.13503*, 2024.

Hanbyul Joo, Tomas Simon, Xulong Li, Hao Liu, Lei Tan, Lin Gui, Sean Banerjee, Timothy Godisart, Bart Nabbe, Iain Matthews, Takeo Kanade, and Yaser Sheikh. Panoptic studio: A massively multiview system for social motion capture. In *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, 2015.

Nikita Karaev, Iurii Makarov, Jianyuan Wang, Natalia Neverova, Andrea Vedaldi, and Christian Rupprecht. Cotracker3: Simpler and better point tracking by pseudo-labelling real videos. In *Proc. arXiv:2410.11831*, 2024.

Tianye Li, Mira Slavcheva, Michael Zollhoefer, Simon Green, Christoph Lassner, Changil Kim, Tanner Schmidt, Steven Lovegrove, Michael Goesele, Richard Newcombe, and Zhaoyang Lv. Neural 3D video synthesis from multi-view video. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 5521–5531, 2022.

Zizhang Li, Hong-Xing Yu, Wei Liu, Yin Yang, Charles Herrmann, Gordon Wetzstein, and Jiajun Wu. Wonderplay: Dynamic 3d scene generation from a single image and actions. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, 2025.

Mingxiang Liao, Hannan Lu, Xinyu Zhang, Fang Wan, Tianyu Wang, Yuzhong Zhao, Wangmeng Zuo, Qixiang Ye, and Jingdong Wang. Evaluation of text-to-video generation models: A dynamics perspective, 2024. URL https://arxiv.org/abs/2407.01094.

Jie Liu, Gongye Liu, Jiajun Liang, Yangguang Li, Jiaheng Liu, Xintao Wang, Pengfei Wan, Di Zhang, and Wanli Ouyang. Flow-grpo: Training flow matching models via online rl, 2025.

Min Liu, Gang Yang, Siyuan Luo, and Lin Shao. Softmac: Differentiable soft body simulation with forecast-based contact model and two-way coupling with articulated rigid bodies and clothes. *arXiv preprint arXiv:2312.03297*, 2023a.

Wei Liu, Ziyu Chen, Zizhang Li, Yue Wang, Hong-Xing Yu, and Jiajun Wu. Realwonder: Real-time physical action-conditioned video generation. *arXiv preprint arXiv:2603.05449*, 2026.

Yaofang Liu, Xiaodong Cun, Xuebo Liu, Xintao Wang, Yong Zhang, Haoxin Chen, Yang Liu, Tieyong Zeng, Raymond Chan, and Ying Shan. Evalcrafter: Benchmarking and evaluating large video generation models, 2024a. URL https://arxiv.org/abs/2310.11440.

Yixin Liu, Kai Zhang, Yuan Li, Zhiling Yan, Chujie Gao, Ruoxi Chen, Zhengqing Yuan, Yue Huang, Hanchi Sun, Jianfeng Gao, Lifang He, and Lichao Sun. Sora: A review on background, technology, limitations, and opportunities of large vision models, 2024b. URL https://arxiv.org/abs/2402.17177.

Yuanxin Liu et al. Fetv: A benchmark for fine-grained evaluation of open-domain text-to-video generation. *arXiv preprint arXiv:2311.01813*, 2023b.

Yunze Liu, Yun Liu, Che Jiang, Kangbo Lyu, Weikang Wan, Hao Shen, Boqiang Liang, Zhoujie Fu, He Wang, and Li Yi. Hoi4d: A 4d egocentric dataset for category-level human-object interaction. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2022.

Haoran Lu, Yitong Li, Ruihai Wu, Chuanruo Ning, Yan Shen, and Hao Dong. UniGarment: A unified simulation and benchmark for garment manipulation. In *ICRA Workshop on Representing and Manipulating Deformable Objects*, 2024a. URL `https://garmentlab.github.io/`. Extended Abstract; Oral.

Haoran Lu, Ruihai Wu, Yitong Li, Sijie Li, Ziyu Zhu, Chuanruo Ning, Yan Shen, Longzan Luo, Yuanpei Chen, and Hao Dong. Garmentlab: A unified simulation and benchmark for garment manipulation. In *Advances in Neural Information Processing Systems*, 2024b.

Jonathon Luiten et al. Dynamic 3d gaussians: Tracking by persistent dynamic view synthesis. *arXiv preprint arXiv:2308.09713*, 2023.

Viktor Makoviychuk, Lukasz Wawrzyniak, Yunrong Guo, Michelle Lu, Kier Storey, Miles Macklin, David Hoeller, Nikita Rudin, Arthur Allshire, Ankur Handa, et al. Isaac gym: High performance gpu-based physics simulation for robot learning. In *Thirty-fifth Conference on Neural Information Processing Systems Datasets and Benchmarks Track*, 2021.

Fanqing Meng, Jiaqi Liao, Xinyu Tan, Wenqi Shao, Quanfeng Lu, Kaipeng Zhang, Cheng Yu, Dianqi Li, Yu Qiao, and Ping Luo. Towards world simulator: Crafting physical commonsense-based benchmark for video generation. In *International Conference on Machine Learning*, 2025.

M. Mittal et al. Isaac lab: A gpu-accelerated simulation framework for multi-modal robot learning. *arXiv preprint arXiv:2511.04831*, 2025.

Saman Motamed, Laura Culp, Kevin Swersky, Priyank Jaini, and Robert Geirhos. Do generative video models understand physical principles? *arXiv preprint arXiv:2501.09038*, 2025.

Chong Mou et al. T2i-adapter: Learning adapters to dig out more controllable ability for text-to-image diffusion models. *arXiv preprint arXiv:2302.08453*, 2023.

Matthias Müller, Bruno Heidelberger, Marcus Hennix, and John Ratcliff. Position based dynamics. *Journal of Visual Communication and Image Representation*, 18(2):109–118, 2007.

NVIDIA. Cosmos 3: Omnimodal world models for physical ai. *arXiv preprint arXiv:2606.02800*, 2026.

NVIDIA Corporation. Isaac sim: A robotics simulation and synthetic data generation platform. `https://developer.nvidia.com/isaac/sim`, 2024.

NVIDIA Isaac Lab. Tiled rendering in isaac lab. `https://docs.nvidia.com/learning/physical-ai/getting-started-with-isaac-lab/latest/an-introduction-to-robot-learning-and-isaac-lab/03-available-robots-and-environments/03-tiled-rendering.html`, 2026. Accessed: 2026-01.

Keunhong Park, Utkarsh Sinha, Peter Hedman, Jonathan T Barron, Sofien Bouaziz, Dan B Goldman, Ricardo Martin-Brualla, and Steven M Seitz. HyperNeRF: A higher-dimensional representation for topologically varying neural radiance fields. *ACM Transactions on Graphics*, 40(6), 2021.

Adam Polyak, Amit Zohar, Andrew Brown, Andros Tjandra, Animesh Sinha, Ann Lee, Apoorv Vyas, Bowen Shi, Chih-Yao Ma, Ching-Yao Chuang, et al. Movie gen: A cast of media foundation models. *arXiv preprint arXiv:2410.13720*, 2024.

Albert Pumarola, Enric Corona, Gerard Pons-Moll, and Francesc Moreno-Noguer. D-NeRF: Neural radiance fields for dynamic scenes. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 10318–10327, 2021.

Tim Rädsch, Yuki M. Asano, Hilde Kuehne, Stefan Bauer, Priyank Jaini, Robert Geirhos, and Carsten T. Lüth. Physics-iq verified. *arXiv preprint arXiv:2606.18943*, 2026.

Marc Rigter et al. Avid: Adapting video diffusion models to world models. *arXiv preprint arXiv:2410.12822*, 2024.

Sand.ai. MAGI-1: Autoregressive video generation at scale. *arXiv preprint arXiv:2505.13211*, 2025.

Christoph Schuhmann et al. Laion-5b: An open large-scale dataset for training next generation image-text models. *arXiv preprint arXiv:2210.08402*, 2022.

Edgar Sucar et al. Dynamic point maps: A versatile representation for dynamic 3d reconstruction. *arXiv preprint arXiv:2503.16318*, 2025.

Kaiyue Sun, Kaiyi Huang, Xian Liu, Yue Wu, Zihan Xu, Zhenguo Li, and Xihui Liu. T2v-compbench: A comprehensive benchmark for compositional text-to-video generation, 2025. URL `https://arxiv.org/abs/2407.14505`.

Zachary Teed and Jia Deng. Raft: Recurrent all-pairs field transforms for optical flow. *ArXiv*, abs/2003.12039, 2020.

Tencent Hunyuan Foundation Model Team. Hunyuanvideo: A systematic framework for large video generative models. *arXiv preprint arXiv:2412.03603*, 2024.

Josh Tobin, Rachel Fong, Alex Ray, Jonas Schneider, Wojciech Zaremba, and Pieter Abbeel. Domain randomization for transferring deep neural networks from simulation to the real world. In *2017 IEEE/RSJ international conference on intelligent robots and systems (IROS)*, pp. 23–30. IEEE, 2017.

Bram Wallace, Meihua Dang, Rafael Rafailov, Linqi Zhou, Aaron Lou, Senthil Purushwalkam, Stefano Ermon, Caiming Xiong, Shafiq Joty, and Nikhil Naik. Diffusion model alignment using direct preference optimization, 2023.

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, Mengyang Feng, Ningyi Zhang, Pandeng Li, Pingyu Wu, Ruihang Chu, Ruili Feng, Shiwei Zhang, Siyang Sun, Tao Fang, Tianxing Wang, Tianyi Gui, Tingyu Weng, Tong Shen, Wei Lin, Wei Wang, Wei Wang, Wenmeng Zhou, Wente Wang, Wenting Shen, Wenyuan Yu, Xianzhong Shi, Xiaoming Huang, Xin Xu, Yan Kou, Yangyu Lv, Yifei Li, Yijing Liu, Yiming Wang, Yingya Zhang, Yitong Huang, Yong Li, You Wu, Yu Liu, Yulin Pan, Yun Zheng, Yuntao Hong, Yupeng Shi, Yutong Feng, Zeyinzi Jiang, Zhen Han, Zhi-Fan Wu, and Ziyu Liu. Wan: Open and advanced large-scale video generative models, 2025.

Chen Wang, Chuhao Chen, Yiming Huang, Zhiyang Dou, Yuan Liu, Jiatao Gu, and Lingjie Liu. Physctrl: Generative physics for controllable and physics-grounded video generation. *arXiv preprint arXiv:2509.20358*, 2025.

Yihan Wang, Lahav Lipson, and Jia Deng. Sea-raft: Simple, efficient, accurate raft for optical flow. *arXiv preprint arXiv:2405.14793*, 2024.

Guanjun Wu, Taoran Yi, Jiemin Fang, Lingxi Xie, Xiaopeng Zhang, Wei Wei, Wenyu Liu, Qi Tian, and Xinggang Wang. 4d gaussian splatting for real-time dynamic scene rendering. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 20310–20320, 2024a.

Rundi Wu et al. Cat4d: Create anything in 4d with multi-view video diffusion models. *arXiv preprint arXiv:2411.18613*, 2024b.

Tianyi Xie, Zeshun Zong, Yuxing Qiu, Xuan Li, Yutao Feng, Yin Yang, and Chenfanfu Jiang. Physgaussian: Physics-integrated 3d gaussians for generative dynamics. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2024.

Tianshuo Xu, Zhifei Chen, Leyi Wu, Hao Lu, and Ying cong Chen. Motion forcing: A decoupled framework for robust video generation in motion dynamics. *arXiv preprint arXiv:2603.10408*, 2026.

Lihe Yang, Bingyi Kang, Zilong Huang, Zhen Zhao, Xiaogang Xu, Jiashi Feng, and Hengshuang Zhao. Depth anything v2. *arXiv:2406.09414*, 2024a.

Lihe Yang et al. Depth anything: Unleashing the power of large-scale unlabeled data. *arXiv preprint arXiv:2401.10891*, 2024b.

Zhuoyi Yang, Jiayan Teng, Wendi Zheng, Ming Ding, Shiyu Huang, Jiazheng Xu, Yuanming Yang, Wenyi Hong, Xiaohan Zhang, Guanyu Feng, Da Yin, Yuxuan Zhang, Weihan Wang, Yean Cheng, Bin Xu, Xiaotao Gu, Yuxiao Dong, and Jie Tang. Cogvideox: Text-to-video diffusion models with an expert transformer, 2025.

Hong-Xing Yu, Haoyi Duan, Charles Herrmann, William T. Freeman, and Jiajun Wu. Wonderworld: Interactive 3d scene generation from a single image. *arXiv preprint arXiv:2406.09394*, 2024a.

Hong-Xing Yu, Haoyi Duan, Junhwa Hur, Kyle Sargent, Michael Rubinstein, William T. Freeman, Forrester Cole, Deqing Sun, Noah Snavely, Jiajun Wu, and Charles Herrmann. Wonderjourney: Going from anywhere to everywhere. *2024 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 6658–6667, 2024b.

Jiwen Yu et al. Gamefactory: Creating new games with generative interactive videos. *arXiv preprint arXiv:2501.08325*, 2025.

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

Jiahao Zhan, Zizhang Li, Hong-Xing Yu, and Jiajun Wu. PerpetualWonder: Long-horizon action-conditioned 4D scene generation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2026. arXiv preprint arXiv:2602.04876.

Ke Zhang et al. Think before you diffuse: Infusing physical rules into video diffusion. *arXiv preprint arXiv:2505.21653*, 2025a.

Lvmin Zhang, Anyi Rao, and Maneesh Agrawala. Adding conditional control to text-to-image diffusion models. *arXiv preprint arXiv:2302.05543*, 2023.

Qiyuan Zhang, Biao Gong, Shuai Tan, Zheng Zhang, Yujun Shen, Xing Zhu, Yuyuan Li, Kelu Yao, Chunhua Shen, and Changqing Zou. Physrvg: Physics-aware unified reinforcement learning for video generative models. *arXiv preprint arXiv:2601.11087*, 2026.

Xiangdong Zhang et al. Videorepa: Learning physics for video generation through relational alignment with foundation models. *arXiv preprint arXiv:2505.23656*, 2025b.

Haoyu Zhen, Qiao Sun, Hongxin Zhang, Junyan Li, Siyuan Zhou, Yilun Du, and Chuang Gan. Tesseract: learning 4d embodied world models. *arXiv preprint arXiv:2504.20995*, 2025.

Dian Zheng, Ziqi Huang, Hongbo Liu, Kai Zou, Yinan He, Fan Zhang, Yuanhan Zhang, Jingwen He, Wei-Shi Zheng, Yu Qiao, and Ziwei Liu. Vbench-2.0: Advancing video generation benchmark suite for intrinsic faithfulness. *arXiv preprint arXiv:2503.21755*, 2025.

Zangwei Zheng, Xiangyu Peng, Tianji Yang, Chenhui Shen, Shenggui Li, Hongxin Liu, Yukun Zhou, Tianyi Li, and Yang You. Open-sora: Democratizing efficient video production for all, 2024.

APPENDIX OVERVIEW

This appendix provides supplementary materials that support the main paper and improve clarity, completeness, and reproducibility. Specifically, it expands on data construction, training details, extended experiments, qualitative analyses, theoretical details, discussion of limitations and impact, and physics-based simulation settings that are not fully described in the main text due to space constraints. To further support reproducibility and future research, we will release the code, datasets, introduction slides, and video demos through the project page: `https://sensational-brioche-7657e7.netlify.app/`

Reproducibility, assets, and release. The supplementary material documents our data construction pipeline, benchmark protocol, model components, and stage-wise training procedure. For anonymous submission, we provide an anonymized project page for paper resources and plan to release code, benchmark assets, and representative data artifacts under a CC BY 4.0 license upon acceptance, together with documentation for reproduction and responsible use.

Training summary. Detailed stage-wise training protocol, compute cost, and inference overhead are collected in Appendix D. Figure 4 previews a Phys4D-generated scene.



Figure 4: **Qualitative result of a Phys4D-generated scene.**

Appendix A: Data. We describe the data sources used throughout training, including internet videos, model-generated samples, and simulation-generated data. This section details data preprocessing, filtering strategies, and annotation procedures for depth and motion supervision, the simulation data hierarchy (templates, physical configurations, environments, videos) with its template-level held-out split, and capture settings, clarifying how diverse supervision signals are constructed across different stages of training.

Appendix B: Detailed Problem Formulation. This section expands the RGB-D-motion interface used throughout the paper, clarifying how depth, optical flow, lifted 3D motion, and trajectories define a video-compatible representation for physics-consistent world modeling.

Appendix C: Implementation. This appendix provides implementation details for the proposed **Phys4D** framework, covering data preparation, model training, and reinforcement learning fine-tuning. Our implementation follows the three-stage training paradigm described in the main paper and is designed to progressively augment video diffusion models from appearance-level generation with a physics-aware **RGB-D-motion interface**.

Appendix D: Training Details. This section summarizes the stage-wise training protocol, hardware footprint, the measured inference overhead introduced by Phys4D, and the Stage-III policy-optimization details, consolidating reproducibility details in one place.

Appendix E: Qualitative Results. We present additional qualitative results to complement the quantitative evaluations in the main paper. This section includes visual demonstrations of generated videos, depth maps, and motion fields, highlighting improvements in fine-grained physical consistency, temporal coherence, and sequence-level trajectory coherence.

Appendix F: Experiments. We report full per-metric results for the external physics benchmarks, Physics-IQ Original and Verified protocol details, a unified comparison with RealWonder, TesserAct, and PhysGen3D, long-video results, and visual-prior preservation analyses.

Appendix G: Additional Ablations for RQ4. This section collects a unified ablation table and supporting analyses of training stages, loss components, auxiliary heads, and local temporal tolerance. It also documents the simulator-video RGB flow-matching control used in Sec. 6.4.

Appendix H: Additional 4D World-Level Diagnostics. This section presents RQ2 diagnostics on the RGB-D-motion world interface, including per-frame geometry, local temporal consistency, one complete seven-method 4D table, and rigid-body analysis on held-out simulated scenes. Worldline and rigid-body trajectory diagnostics use a shared frozen CoTracker3 readout of generated RGB videos, with simulator trajectories as references.

Appendix I: Theory of Stage III RL. We include mathematical preliminaries and theoretical details underlying our approach, including flow matching, deterministic ODE sampling, and the stochastic formulations used for reinforcement learning fine-tuning. This section contains derivations and clarifications that support the methodology presented in the main paper.

Appendix J: Extended Related Work. We discuss related work that could not be fully covered in the main paper, providing further context on physics-aware video generation, 4D modeling, and learning-based world representations.

Appendix K: Physics Simulation. We describe the physics simulation settings used for supervision and reward construction, including scene configurations, object properties, and motion extraction procedures. This section explains how simulation is leveraged as a source of high-fidelity geometric and motion information for fine-grained physical alignment.

Appendix L: Limitations and Impact Statement. This section consolidates the paper's limitations, including the dependence on simulator-paired rollouts, the scope of the 4D alignment reward, the scale of the RGB-D-motion interface, and the generalization scope of the evaluation, together with the broader impact and responsible release stance.

A Data

A.1 Simulation Data

To enable large-scale 4D data collection at high throughput and low cost, we build our simulation pipeline on an *asynchronous parallel execution* framework. Our system collects data from a large number of simulation sub-environments concurrently, significantly improving GPU utilization and simulation throughput compared to sequential or synchronously parallel simulators. This design allows us to generate large-scale, diverse simulation data at substantially higher speed and lower computational cost.

Tiled Rendering for Efficient Visual Simulation. To further improve rendering efficiency, we adopt *tiled camera rendering*, where multiple camera views from different simulation sub-environments are rendered within a single large framebuffer. This approach amortizes rendering overhead across environments and avoids redundant GPU context switching. We follow the tiled rendering strategy provided by Isaac Lab (NVIDIA Isaac Lab, 2026), which enables scalable visual data generation with minimal performance degradation as the number of parallel environments increases.

Asynchronous Execution. While parallelism is critical for simulator efficiency, naive synchronous execution leads to significant idle time when sub-environments exhibit heterogeneous dynamics. In vision-based control and data collection, a policy typically performs inference once to generate a target pose or waypoint, after which the robot executes a trajectory of variable length. Synchronizing such trajectories across environments results in unnecessary blocking and under-utilization of computational resources.

To address this challenge, our simulator supports *fully asynchronous execution*. Each policy-driven motion is encapsulated as an atomic action with a well-defined interface, allowing high-level policies

to invoke and monitor actions independently across environments. This design eliminates global synchronization barriers and maximizes hardware utilization, enabling efficient closed-loop data collection under heterogeneous execution.

Data Management and Asynchronous Control. Supporting asynchronous execution is non-trivial, as the underlying simulator operates in a SIMD (Single Instruction, Multiple Data) paradigm, whereas asynchronous execution requires MIMD (Multiple Instruction, Multiple Data) semantics. To bridge this gap, we introduce a motion-planning-based abstraction layer that translates high-level asynchronous commands into low-level, time-continuous trajectories. This abstraction allows each sub-environment to progress independently while preserving simulator stability.

Scene Management and Heterogeneous Environments. To support heterogeneous parallel simulation, different sub-environments may require distinct scene configurations. We address this by introducing a *Layout Manager* that programmatically controls asset placement across environments. The Layout Manager imports user-specified objects with configurable randomization, enabling diverse, controllable, and reproducible scene layouts. This design allows us to efficiently scale data collection across a wide range of scene configurations within a single simulator instance.

Multi-Modal Camera Strategies. We employ two complementary camera strategies to balance static scene coverage and dynamic viewpoint variation. **Fixed multi-view cameras** provide dense spatial coverage by placing multiple static cameras around each scene, capturing aligned RGB images together with calibrated intrinsics and extrinsics and dense geometric annotations such as depth and optical flow. In parallel, **dynamic trajectory cameras** simulate realistic cinematography by sampling continuous camera motions (e.g., orbit, dolly, crane, tracking, and first-person) along parameterized trajectories, producing frame-level pose annotations for training trajectory-aware video generation models.

Simulation Throughput and Data Scale. Our data collection pipeline is optimized for high-throughput simulation. On a single NVIDIA RTX 4090 GPU with 24 GB memory, we run up to 1,024 simulation environments in parallel, enabling efficient large-scale data generation at relatively low computational cost. Each environment is simulated for 10 seconds of effective interaction time.

We construct a base set of 200 simulation *templates* and expand it to approximately 250,000 distinct environments through domain randomization over initial object configurations, physical parameters, and surface textures. For each environment, we collect five camera trajectories, including three fixed-view cameras and two dynamic camera motions, providing diverse spatial and temporal observations of the same underlying physical scene. This yields approximately 1.25M annotated 10-second videos, corresponding to about 3.5K hours of simulation video ($1.25\mathrm{M} \times 10\,\mathrm{s} \approx 3.5\mathrm{K}\,\mathrm{h}$), together with over 15 TB of multi-modal annotations.

Data hierarchy. The quantities above refer to different levels of one hierarchy, summarized in Table 5. A *template* is a parameterized physical-interaction scene (object categories, solvers, interaction type, and admissible parameter ranges), not a fixed video or trajectory. A *physical configuration* is one combination of the material, environmental, and geometric attributes of Table 25; the 200 templates yield more than 20,000 unique physical configurations. An *environment* additionally instantiates a physical configuration under a particular set of initial object meshes and layouts and a surface-texture/visual variant, giving approximately 250K environments. Finally, each environment is rolled out and recorded by five cameras, giving 1.25M videos with frame-aligned RGB, depth, optical flow, dynamic and object masks, visibility, camera poses, and 3D point trajectories. These five camera recordings share one underlying physical evolution and must not be counted as five independent physical trials. The total corpus size also does not identify the number of samples actually consumed by each training stage or used for model selection.

Held-out split. We split *before* rollout generation at the template level: 180 templates are used for training and 20 templates are held out (a 90/10 split), and all environments and rollouts derived from a held-out template are excluded from Stage II and Stage III training. The held-out scenes used for the 4D diagnostics in Appendix H are drawn exclusively from the 20 held-out templates. Asset pools, object categories, solvers, and physical-parameter ranges may overlap between the two splits by design: the evaluation tests generalization to unseen templates, initial states, and configurations

Table 5: **Simulation data hierarchy and held-out split.** Each level instantiates the level above it; the held-out split is made at the template level before any rollout is generated.

Level	Count	Definition	Split
Templates	200	parameterized interaction scenes (categories, solvers, parameter ranges)	180 train / 20 held-out
Physical configurations	>20K	combinations of material, environment, and geometry attributes (Table 25)	inherited from template
Environments	≈250K	configuration × initial meshes/layout × texture/visual variant	inherited from template
Videos	1.25M	five 10-second camera rollouts per environment (3 fixed, 2 dynamic)	inherited from template

within the supported physical regimes, not to unseen solvers or out-of-distribution physics. A split manifest listing the template identifiers of each split will be released with the data.

Capture settings. All modalities are exported at 30 Hz and are frame-aligned. Cameras use wide field-of-view, Kinect-like intrinsics (about $80^\circ \times 51^\circ$) at large stand-off distances; dynamic-camera rollouts follow slow cinematography-style trajectories (orbit, lateral track, crane, dolly, first-person).

A.2 Internet Video Pretraining Data

Motivation. To improve the robustness and generalization of the depth and motion heads, we leverage a curated set of real-world videos, drawn from a large crawled candidate pool, for pseudo-supervised pretraining. Compared to simulation data, Internet videos exhibit substantially greater diversity in scene layout, appearance, and motion patterns, which is critical for learning domain-agnostic geometry and motion representations. This pretraining stage equips the model with strong visual and geometric priors before introducing physics-grounded supervision.

Data Sources. We assemble the pretraining corpus using a hybrid collection strategy that combines automated web crawling with selective manual curation. Videos are sourced from multiple public platforms, including YouTube, Pexels, Pixabay, Mixkit, and Bilibili, ensuring broad coverage of real-world environments and camera motions. We retain only videos with permissive licenses when license metadata is available, exclude private or personal content, and provide a removal mechanism for released metadata.

Automated Quality Filtering. To maintain data quality at scale, we apply a multi-stage automated filtering pipeline. We first use the LAION aesthetic scorer (Schuhmann et al., 2022) to remove visually low-quality or noisy clips. Next, we employ an OCR-based text detector based on PaddleOCR to identify and discard videos containing prominent overlaid text, watermarks, or subtitles that may interfere with depth and motion estimation. Finally, videos that are extremely short, corrupted, or visually inconsistent are automatically removed. This filtering process yields a curated, diverse, and high-quality video set suitable for robust geometry and motion pretraining.

Final Dataset Curation. Starting from an initial pool of approximately 100,000 candidate video clips, our automated filtering pipeline retains about 5,000 high-quality videos that satisfy our fine-grained requirements. These selected clips exhibit sufficient motion diversity, clear object boundaries, stable visual quality, and appropriate scene and object scales, making them suitable for robust depth and motion pretraining. The curation is scale-aware and deliberately strict: clips with ambiguous or negligible motion, severe occlusion, unreliable or degenerate geometry (for example, large textureless, transparent, or strongly reflective regions), and low or unstable estimator confidence are removed. This curated subset forms the final Internet video dataset used in our pseudo-supervised pretraining stage.

Role of Stage-I pseudo-labels and noise control. The curated 5K-clip set is used only for Stage I, together with videos self-generated by the base model; it is not the source of the generator's real-video prior, which comes from the backbone's own pretraining. Stage I trains only the lightweight depth and motion heads while the RGB generator is frozen, so pseudo-label noise cannot alter the pretrained RGB generator; it affects only the initialization of the interface. Pseudo-label noise is further controlled by estimator confidence masks, robust regression losses, and per-clip depth normalization (Appendix C.2), and the heads are subsequently recalibrated in Stage II with simulator metric ground truth for depth, motion, visibility, dynamic masks, and warp consistency, which directly covers fluids, deformables, and occlusions, where off-the-shelf estimators are least reliable. Our claim is therefore

not that off-the-shelf pseudo-labels are noise-free, but that carefully curated real-video supervision provides an RGB-aligned 4D initialization that is subsequently corrected and physically grounded by simulation.

B Detailed Problem Formulation

In this section, we provide additional details for the RGB-D-motion formulation used in Sec. 3. The goal is not to define a complete dynamic 3D reconstruction problem, but to specify a video-compatible interface through which physical structure can be exposed, supervised, and evaluated.

B.1 RGB-D-motion Interface

Given the input image I_0 and an optional text prompt p, the model generates RGB frames $\{I_t\}_{t=0}^{T}$ together with geometry and motion variables:

$$\{D_t\}_{t=0}^{T}, \qquad \{F_{t\to t+1}\}_{t=0}^{T-1}.$$

Here $D_t \in \mathbb{R}^{H\times W}$ denotes a depth map in the camera coordinate system, and $F_{t\to t+1} \in \mathbb{R}^{H\times W\times 2}$ denotes optical flow from frame t to frame $t+1$. These variables form the minimal interface used by Phys4D: RGB frames provide visual observations, depth exposes per-frame geometry, and flow exposes inter-frame motion.

Throughout the paper, we use "4D" to refer to geometry and motion evolving over time. This is weaker than full dynamic volumetric reconstruction, but stronger than purely 2D video generation. It provides a tractable representation that is compatible with video diffusion models and can be supervised using both pseudo-labels from real videos and structural annotations from simulation.

B.2 Lifting Depth and Flow to 3D Motion

The RGB-D-motion interface can be lifted into camera-space 3D motion when camera intrinsics are available or estimated. Let $u = (x, y)$ be a pixel coordinate and $\tilde{u} = (x, y, 1)^\top$ its homogeneous coordinate. Given camera intrinsics K, the corresponding 3D point at time t is

$$X_t(u) = D_t(u)K^{-1}\tilde{u}.$$

The optical flow predicts a correspondence

$$u' = u + F_{t\to t+1}(u).$$

Using the depth at the target location, the corresponding point at time $t+1$ is

$$X_{t+1}(u') = D_{t+1}(u')K^{-1}\tilde{u}'.$$

The induced scene flow is therefore

$$S_{t\to t+1}(u) = X_{t+1}(u') - X_t(u).$$

This relation couples depth and 2D motion into a single 3D displacement field. In practice, this lifting is only evaluated on valid correspondences, excluding occluded pixels, out-of-bound flow endpoints, and regions with unreliable depth or motion.

B.3 Trajectory Representation

Beyond adjacent-frame motion, physical consistency often depends on longer-range evolution. We therefore define lifted trajectories by recursively composing the predicted flow and depth over time. Starting from a pixel u_0 in frame 0, its image trajectory is

$$u_{t+1} = u_t + F_{t\to t+1}(u_t),$$

and its lifted 3D trajectory is

$$\tau(u_0) = \{X_t(u_t)\}_{t=0}^{T}, \qquad X_t(u_t) = D_t(u_t)K^{-1}\tilde{u}_t.$$

The collection of valid lifted trajectories is denoted as

$$\mathcal{T} = \{\tau(u_0)\}_{u_0\in\Omega_{\text{valid}}},$$

where Ω_{valid} contains pixels whose trajectories remain visible and geometrically valid for the considered time horizon. These trajectories summarize non-local scene evolution and are used to reason about physical behavior that cannot be fully captured by per-frame depth or adjacent-frame flow alone.

B.4 Simulation-derived Structural Supervision

The coupled-physics simulator provides temporally aligned structural annotations

$$\mathcal{Y}^{\text{sim}}_{\text{structure}} = \{D, F, S, M, V, C, \mathcal{T}\}^{\text{sim}},$$

where D is depth, F is optical flow, S is scene flow, M is a dynamic or object mask, V is visibility, C is contact information, and $\mathcal{T}$ denotes lifted trajectories. These annotations are exported from simulation because they are structural and physically meaningful. In contrast, rendered RGB frames from simulation are treated as domain-specific observations and are not used as the primary appearance target for training the video generator.

This distinction is important. The simulator provides executable physical knowledge: geometry, motion, contact, deformation, coupling, and visibility. The pretrained video diffusion model provides the real-video visual prior. Phys4D transfers the former into the latter through structural variables under appearance-constrained adaptation, rather than by fully steering the generator toward simulator-specific appearance statistics.

B.5 Geometry-motion Consistency

Flow-based warping establishes correspondences, but a corresponding surface point need not retain the same scalar depth when the object or camera moves axially. For simulation-paired training, let $\mathcal{W}_t$ transport depth from frame t to the frame-$t+1$ grid using fixed simulator flow. On valid correspondences, we match the signed predicted depth change to the simulator depth change:

$$\mathcal{W}_t(\hat{D}_t) - \hat{D}_{t+1} \approx \mathcal{W}_t(D^{\text{sim}}_t) - D^{\text{sim}}_{t+1}.$$

This supervises depth evolution without imposing scalar-depth invariance. It complements the 3D lifting relation above, which uses target-frame depth to construct scene flow. The geometry-warp loss does not use simulator RGB as a photometric target.

B.6 Scope of the Formulation

The formulation makes three design choices. First, it uses depth, optical flow, scene flow, and trajectories as a tractable proxy for 4D scene evolution, rather than requiring full dynamic volumetric reconstruction. Second, it separates visual realism from physical structure: visual realism is inherited from the pretrained video model, while physical structure is injected through geometry, motion, visibility, contact, and trajectory supervision. Third, it supports both local and global physical constraints: local consistency is expressed through depth-flow-scene-flow relations, while global consistency is expressed through lifted trajectories over the generated sequence.

C Implementation

C.1 Stage-I Auxiliary Head Architecture

Stage I augments the frozen pretrained DiT video backbone with two lightweight auxiliary heads for geometry and motion prediction. The backbone parameters are kept fixed throughout this stage; only the auxiliary heads are optimized.

Motion head. We attach a motion prediction head to diffusion features captured from the DiT output layer. The head consists of a LayerNorm followed by a modulated linear projection, producing patchified motion tokens that are unpatchified into a multi-channel motion field:

$$\hat{M}_{t\to t+1} = (\hat{dx}, \hat{dy}, \hat{c}_2, \hat{c}_3) \in \mathbb{R}^{H\times W\times 4}.$$

The first two channels $(\hat{dx}, \hat{dy})$ represent the optical flow $\hat{F}_{t\to t+1}$ between adjacent frames, while the remaining channels encode auxiliary motion information used by the motion head. Unless otherwise specified, the supervised motion loss is applied to the optical-flow channels.

Depth head. We attach a depth prediction head following the spatiotemporal design of Video Depth Anything (Chen et al., 2025b). The head follows a DPT-style decoder and incorporates temporal attention and multi-scale feature fusion to encourage temporally coherent depth prediction. Specifically, we uniformly sample four feature maps from the DiT backbone and process them with Reassemble and Fusion layers to form a multi-scale feature pyramid. To enable cross-frame reasoning, four temporal attention layers are inserted at low-resolution stages, where each layer applies multi-head self-attention along the temporal dimension with absolute positional embeddings. The decoder outputs per-frame depth maps

$$\hat{D}_t \in \mathbb{R}^{H \times W}.$$

Both heads share the same frozen diffusion feature interface. This design exposes depth and motion variables required by later physics supervision, while leaving the pretrained RGB video generator unchanged.

C.2 Stage-I Pseudo-Supervised Losses

Stage I trains the auxiliary heads using pseudo labels from curated internet videos and videos self-generated by the pretrained generator. We annotate both sources with off-the-shelf monocular depth (Chen et al., 2025b) and optical flow (Teed & Deng, 2020) estimators, obtaining pseudo depth $\tilde{D}_t$ and pseudo flow $\tilde{F}_{t\to t+1}$. Estimator confidence, validity, or visibility maps are used as supervision weights when available.

Depth loss. Since monocular depth is scale ambiguous, we normalize both the predicted and pseudo depth within each clip before supervision. Let $\mathcal{N}(\cdot)$ denote per-clip depth normalization. The pseudo-depth loss is

$$\mathcal{L}_D^{pseudo} = \sum_{t,u} w_t^D(u)\, \rho\left(\mathcal{N}(\hat{D}_t(u)) - \mathcal{N}(\tilde{D}_t(u))\right),$$

where u indexes image locations, $w_t^D(u)$ is the depth confidence or validity weight, and $\rho(\cdot)$ is a robust regression loss such as smooth-ℓ_1.

Motion loss. The motion head is supervised using pseudo optical flow between adjacent frames:

$$\mathcal{L}_F^{pseudo} = \sum_{t,u} w_t^F(u)\, \rho\left(\hat{F}_{t\to t+1}(u) - \tilde{F}_{t\to t+1}(u)\right),$$

where $w_t^F(u)$ masks invalid, low-confidence, or occluded correspondences.

Stage-I objective. The total Stage-I loss is

$$\mathcal{L}_{S1} = \lambda_D \mathcal{L}_D^{pseudo} + \lambda_F \mathcal{L}_F^{pseudo}.$$

Gradients from $\mathcal{L}_{S1}$ are applied only to the depth and motion heads. The pretrained DiT backbone and the base video generation objective remain unchanged. Thus, Stage I provides a prior-preserving RGB-D-motion interface without injecting simulation physics or altering the original RGB generator.

C.3 Stage-II Masked Geometry-Motion Losses

Stage II uses simulation annotations to inject local physical structure while keeping the pretrained video backbone frozen. Let θ denote the frozen backbone parameters, ϕ the trainable physics adapters, ψ the learned dynamic gate, and η_D, η_F the depth and motion heads. Gradients from all Stage-II losses are blocked from θ and update only $\{\phi, \psi, \eta_D, \eta_F\}$.

Table 6 summarizes the supervision source, updated modules, and role of each Stage-II term. Importantly, the RGB-level term is a self-consistency regularizer on generated frames and does not use simulator-rendered RGB as a photometric target, whereas $\mathcal{L}_{FM}^{high}$ does use latents encoded from simulator-rendered RGB videos under a restricted high-noise adapter-only pathway.

Table 6: **Stage-II loss summary.** Stage II combines restricted simulator-latent adaptation with structure-level supervision. $\mathcal{L}_{FM}^{high}$ uses latents encoded from simulator-rendered RGB videos, but only at high-noise denoising steps and only through LoRA adapters. In contrast, $\mathcal{L}_{rgb_warp}^{self}$ compares generated RGB frames under motion and does not match generated frames to simulator-rendered RGB.

Loss	Uses simulator RGB latent?	Photometric sim-RGB reconstruction target?	Supervision source	Updated modules	Purpose
$\mathcal{L}_{FM}^{high}$	Yes, simulator-rendered RGB latent	No, not a low-noise photometric reconstruction loss	high-noise noised training latents encoded from simulator-rendered RGB videos	adapters only	keep adapted high-noise denoising on the video manifold
$\mathcal{L}_{depth}^{mask}$	No	No	sim depth + visibility + dynamic masks	depth head + adapters	metric local geometry supervision
$\mathcal{L}_{motion}^{mask}$	No	No	sim flow / projected motion + visibility + dynamic masks	motion head + adapters	local physical motion supervision
$\mathcal{L}_{geom_warp}^{mask}$	No	No	sim depth at both frames + sim flow + visibility + dynamic masks	depth head + adapters	signed depth-change matching along simulator correspondences
$\mathcal{L}_{rgb_warp}^{self}$	No	No	generated RGB frames + predicted motion	adapters + motion head	temporal RGB self-consistency without simulator RGB matching
$\mathcal{L}_{gate}$	No	No	sim dynamic / object masks	gate + adapters	learn physics-relevant regions

Mask weighting. For each frame t, the simulator provides a dynamic or object mask M_t^{dyn} and visibility masks M_t^{vis} or $M_{t\to t+1}^{vis}$. We use dynamic masks as soft weights rather than hard exclusions:

$$w_t(u) = 1 + \beta M_t^{dyn}(u), \tag{8}$$

where u indexes image locations and β controls the emphasis on physics-relevant regions. This preserves supervision on global scene geometry while increasing the contribution of moving objects, deformations, and interaction regions.

Masked depth supervision. Given simulator depth D_t^{sim} and predicted depth $\hat{D}_t$, we define

$$\mathcal{L}_{depth}^{mask} = \frac{\sum_{t,u} M_t^{vis}(u)\, w_t(u)\, \rho_D\left(\hat{D}_t(u) - D_t^{sim}(u)\right)}{\sum_{t,u} M_t^{vis}(u)\, w_t(u) + \epsilon}, \tag{9}$$

where ρ_D is a robust regression loss, e.g., smooth-ℓ_1, and ϵ avoids division by zero. Since simulation provides metric depth, we supervise depth in the simulator scale; when necessary, we apply per-sequence normalization for numerical stability.

Depth scale across stages. The Stage-I and Stage-II depth scales are not mixed. Stage I supervises the depth head with per-clip *normalized* pseudo depth (Appendix C.2) only to initialize the interface; Stage II then recalibrates the same head to simulator *metric* depth through Eq. (9), and all lifted 4D quantities used for the geometry-warp term, the Stage-III reward, and the 4D diagnostics (Appendix H) are computed from this metric-calibrated head. Whenever a stabilizing per-sequence normalization is applied, it is applied identically to the generated and the simulator point sets, so the scale of the lifted 4D points is unambiguous within every comparison.

Masked motion supervision. The simulator provides optical flow or projected motion fields $F_{t\to t+1}^{sim}$. We supervise the motion head by

$$\mathcal{L}_{motion}^{mask} = \frac{\sum_{t,u} M_{t\to t+1}^{vis}(u)\, w_t(u)\, \rho_F\left(\hat{F}_{t\to t+1}(u) - F_{t\to t+1}^{sim}(u)\right)}{\sum_{t,u} M_{t\to t+1}^{vis}(u)\, w_t(u) + \epsilon}, \tag{10}$$

where $M^{vis}_{t\to t+1}$ removes invalid or occluded correspondences. This term trains the motion head to predict physically grounded local motion without using simulator RGB appearance as a target.

Gate supervision. The dynamic gate $\hat{G}_t$ is supervised by simulator object or dynamic masks:

$$\mathcal{L}_{gate} = \mathrm{BCE}(\hat{G}_t, M^{dyn}_t) + \mathrm{Dice}(\hat{G}_t, M^{dyn}_t). \tag{11}$$

Simulator masks are used only during training. At inference time, the gate is predicted from model features and no simulator annotation is required.

GT-compensated geometry warp. Depth and motion supervision alone do not guarantee temporal coherence. Let $\mathcal{W}_t(D) = \mathcal{W}(D, F^{sim}_{t\to t+1})$ transport source-frame depth to the frame-$t+1$ grid using fixed simulator flow. We define signed residuals on that output grid:

$$r^{\mathrm{pred}}_t(u) = \mathcal{W}_t(\hat{D}_t)(u) - \hat{D}_{t+1}(u), \qquad r^{\mathrm{sim}}_t(u) = \mathcal{W}_t(D^{sim}_t)(u) - D^{sim}_{t+1}(u).$$

The loss matches the predicted residual to the simulator's depth-change target:

$$\mathcal{L}^{mask}_{geom_warp} = \frac{\sum_{t,u} m_t(u) w_t(u) \left| r^{\mathrm{pred}}_t(u) - r^{\mathrm{sim}}_t(u) \right|}{\sum_{t,u} m_t(u) w_t(u) + \epsilon}. \tag{12}$$

Here m_t excludes occluded, out-of-bounds, unmatched, and invalid-depth locations. Both m_t and w_t are expressed on the warp-output grid; a source-grid dynamic weight from Eq. (8) is transported or reindexed before use. Predicted and simulator depth share the same metric scale and sampling operator. The simulator flow, target residual, mask, and weights are fixed supervision, so this term does not directly train predicted flow. When predictions equal simulator depths, the residual difference is zero even under axial object or camera motion. The loss constrains depth evolution along correspondences, while Eq. (9) anchors absolute depth. It does not require a negligible-motion approximation.

Generated-frame RGB warp self-consistency. We include a weak RGB warp term to stabilize temporal appearance, but it is not a photometric loss against simulator-rendered RGB. Let $\hat{I}_t$ denote generated RGB frames and $\hat{F}_{t\to t+1}$ the predicted flow. We define

$$\mathcal{L}^{self}_{rgb_warp} = \frac{\sum_{t,u} M^{vis}_{t\to t+1}(u)\, w_t(u)\, \rho_I \left(\mathcal{W}\left(\hat{I}_t, \hat{F}_{t\to t+1}\right)(u) - \hat{I}_{t+1}(u) \right)}{\sum_{t,u} M^{vis}_{t\to t+1}(u)\, w_t(u) + \epsilon}. \tag{13}$$

This term only encourages adjacent generated frames to be consistent under the model-predicted motion. It does not compare $\hat{I}_t$ or $\hat{I}_{t+1}$ to any simulator RGB frame, and therefore does not introduce direct simulator appearance fitting.

C.4 High-Noise Flow Matching for Stage-II Adapter Training

Stage II keeps the pretrained video backbone frozen and trains only lightweight physics adapters. The flow-matching term is used to keep the adapted high-noise denoising trajectory compatible with the video generation manifold. It uses latents encoded from simulator-rendered RGB videos, but it is not a full-backbone simulation-domain fine-tuning objective and it is not a low-noise photometric matching loss to simulator-rendered RGB.

Flow matching preliminaries. Let $\mathbf{z}_0 \sim \mathcal{N}(0, I)$ denote noise and $\mathbf{z}_1 \sim q(\mathbf{z}_1 \mid c)$ denote a data latent conditioned on prompt or image condition c. For interpolation time $s \in [0, 1]$, we use the linear probability path

$$\mathbf{z}_s = (1 - s)\mathbf{z}_0 + s\mathbf{z}_1. \tag{14}$$

The corresponding conditional vector field is

$$u_s(\mathbf{z}_s \mid \mathbf{z}_1) = \frac{d\mathbf{z}_s}{ds} = \mathbf{z}_1 - \mathbf{z}_0. \tag{15}$$

A flow model predicts a velocity field $v(\mathbf{z}_s, s, c)$ and is trained by

$$\mathcal{L}_{FM} = \mathbb{E}_{s,\mathbf{z}_0,\mathbf{z}_1,c} \left[\left\| v(\mathbf{z}_s, s, c) - u_s(\mathbf{z}_s \mid \mathbf{z}_1) \right\|_2^2 \right]. \tag{16}$$

Adapter-parameterized velocity. In Stage II, the velocity field is induced by the frozen pretrained model and the trainable physics adapters:

$$v_{\theta,\phi} = v_\theta + \Delta v_\phi, \tag{17}$$

where θ is frozen and ϕ contains only the lightweight adapter parameters. The adapter update is implemented through the mask-gated residual pathway:

$$h'_\ell = h_\ell + \alpha(\sigma_s)\, \hat{G}_\ell \odot A_\ell(h_\ell). \tag{18}$$

Here σ_s is the noise level associated with interpolation time s, $\alpha(\sigma_s)$ is the high-noise gate, and $\hat{G}_\ell$ is the learned dynamic-region gate.

High-noise restriction. We restrict flow-matching updates to structure-forming high-noise steps. Let $\gamma(s)$ denote a high-noise weighting function, e.g.,

$$\gamma(s) = \mathbb{I}[\sigma_s \geq \sigma_h], \tag{19}$$

or a smooth version

$$\gamma(s) = \mathrm{sigmoid}\left(\frac{\sigma_s - \sigma_h}{\tau}\right). \tag{20}$$

The Stage-II high-noise flow-matching term is

$$\mathcal{L}_{FM}^{high} = \mathbb{E}_{s,\mathbf{z}_0,\mathbf{z}_1,c}\left[\gamma(s)\, \|v_{\theta,\phi}(\mathbf{z}_s, s, c) - (\mathbf{z}_1 - \mathbf{z}_0)\|_2^2\right]. \tag{21}$$

Because θ is frozen, gradients from $\mathcal{L}_{FM}^{high}$ update only ϕ. Moreover, the adapter itself is modulated by $\alpha(\sigma_s)$, so low-noise appearance-refinement steps receive no or negligible simulation-induced updates. This keeps simulation influence concentrated on coarse structure and motion rather than texture, lighting, or renderer-specific appearance statistics. Equivalently, $\mathcal{L}_{FM}^{high}$ uses latents encoded from simulator-rendered RGB videos, but only at high-noise denoising steps and only through LoRA adapters. It is therefore a restricted simulator-latent adaptation term, not full simulator RGB fine-tuning or low-noise photometric appearance fitting.

Gradient paths. The frozen backbone provides features and velocity predictions but is not updated. The trainable paths in Stage II are

$$\mathcal{L}_{FM}^{high} \to \phi, \qquad \mathcal{L}_{depth}^{mask}, \mathcal{L}_{motion}^{mask}, \mathcal{L}_{geom_warp}^{mask}, \mathcal{L}_{rgb_warp}^{self} \to \{\phi, \eta_D, \eta_F\}, \qquad \mathcal{L}_{gate} \to \{\phi, \psi\}.$$

Thus, simulation-derived supervision affects generation only through the mask-gated adapter and the auxiliary geometry-motion heads, while the pretrained backbone remains unchanged.

C.5 Stage-II Overall Objective

The full Stage-II objective is

$$\mathcal{L}_{S2} = \mathcal{L}_{FM}^{high} + \lambda_R \mathcal{L}_{rgb_warp}^{self} + \lambda_D \mathcal{L}_{depth}^{mask} + \lambda_F \mathcal{L}_{motion}^{mask} + \lambda_W \mathcal{L}_{geom_warp}^{mask} + \lambda_G \mathcal{L}_{gate}. \tag{22}$$

The flow-matching term keeps the high-noise adapter pathway compatible with the video generator. The depth, motion, and geometry-warp terms inject local structure from simulation. The RGB warp term is only a generated-frame temporal self-consistency regularizer and does not use simulator RGB as a photometric target. The gate term learns where physical changes occur. All terms are optimized with the pretrained backbone frozen, so Stage II performs controlled local physics injection rather than full simulation-domain fine-tuning.

C.6 Stage-III Reward Implementation: Locally Time-Relaxed 4D Alignment

Stage III evaluates a completed generated video against its paired simulator rollout. Visibility and dynamic-region filtering remove invalid or inactive points. The reward compares frame-level 3D geometry in a fixed one-frame temporal neighborhood and has no category-specific contact or material term.

Generated 4D point construction. Given a generated RGB-D-motion sequence $\{I_t, \hat{D}_t, \hat{F}_{t\to t+1}\}_{t=0}^{T-1}$, we construct a spatiotemporal point set from the predicted depth. For each valid pixel $u = (u_x, u_y)$ at frame t, we back-project the depth into 3D:

$$\mathbf{X}_t^{gen}(u) = \hat{D}_t(u)\, K_t^{-1} \begin{bmatrix} u_x \\ u_y \\ 1 \end{bmatrix}, \tag{23}$$

where K_t is the camera intrinsic matrix. When camera extrinsics are available, we transform all points into a common world or reference coordinate frame:

$$\mathbf{X}_t^{gen,w}(u) = R_t^{-1}\left(\mathbf{X}_t^{gen}(u) - \mathbf{t}_t\right). \tag{24}$$

For fixed-camera environments, the camera coordinate system is used as the common reference frame. We then append a discrete timestamp τ_t to obtain a 4D point:

$$p_{t,u}^{gen} = (\mathbf{X}_t^{gen,w}(u), \tau_t). \tag{25}$$

The motion prediction $\hat{F}_{t\to t+1}$ is used to estimate validity, and dynamic-region masks are used to retain active regions. We then sample a fixed number of valid points per frame to avoid any single frame dominating the reward, and collect them over time to form

$$\mathcal{P}^{gen} = \{p_{t,u}^{gen}\}_{t,u}. \tag{26}$$

For the reward, P_i denotes the nonempty 3D spatial coordinates of the valid generated points at video frame i. The depth and motion heads are frozen from the Stage-II checkpoint, and the same spatial scale is used across frames.

Simulation 4D trajectory construction. The simulator provides geometry and motion in a world coordinate system. For each simulation point $\mathbf{X}_{t,j}^{sim,w}$ at frame t, we construct the corresponding 4D point

$$p_{t,j}^{sim} = \left(\mathbf{X}_{t,j}^{sim,w}, \tau_t\right). \tag{27}$$

If the simulator provides rendered depth rather than explicit world points, we apply the same unprojection and camera-to-world transformation as in Eq. (23)–(24). We remove invalid, invisible, out-of-view, or inactive points using simulator visibility and dynamic-region annotations, and sample the same number of points per frame as for the generated sequence. This yields the simulator trajectory point set

$$\mathcal{P}^{sim} = \{p_{t,j}^{sim}\}_{t,j}. \tag{28}$$

Let Q_j be the nonempty 3D coordinates of valid simulator points at frame j. Both P_i and Q_j are expressed in the same reference coordinate frame.

Frame-pair 3D Chamfer and local temporal cost. For generated frame i and simulator frame j, we use the symmetric 3D Chamfer cost

$$C_{ij} = \frac{1}{|P_i|}\sum_{\mathbf{x}\in P_i}\min_{\mathbf{y}\in Q_j}\|\mathbf{x}-\mathbf{y}\|_2^2 + \frac{1}{|Q_j|}\sum_{\mathbf{y}\in Q_j}\min_{\mathbf{x}\in P_i}\|\mathbf{x}-\mathbf{y}\|_2^2. \tag{29}$$

Let τ_i be the timestamp of video frame i on a fixed shared time grid. The search radius is $r = 1$ for every value of the temporal relaxation parameter λ_t:

$$c_{\lambda_t}(i,j) = \begin{cases} C_{ij} + ((\tau_i - \tau_j)/\lambda_t)^2, & \lambda_t > 0,\ |i-j| \le 1, \\ C_{ii}, & \lambda_t = 0,\ i = j, \\ +\infty, & \text{otherwise.} \end{cases} \tag{30}$$

At the first and last frames, the candidate set is clipped to available indices. Increasing λ_t reduces the cost of an allowed neighboring match but never admits a two-frame or larger offset. At $\lambda_t = 0$, only the same-index frame is feasible; this is an exact branch, not an ϵ approximation.

Bidirectional sequence alignment and reward. For N video frames, we aggregate one best local match per frame in each direction, then map the nonnegative discrepancy to a bounded terminal reward:

$$\mathcal{D}_{\text{align}}^{(1,\lambda_t)}(P,Q) = \frac{1}{2N}\left[\sum_{i=0}^{N-1}\min_{0\le j<N} c_{\lambda_t}(i,j) + \sum_{j=0}^{N-1}\min_{0\le i<N} c_{\lambda_t}(i,j)\right], \tag{31}$$

$$R_{4D}(V;s) = 1 - 2\,\text{clip}\left(\frac{\mathcal{D}_{\text{align}}^{(1,\lambda_t)}(P,Q)}{d_{\text{cap}}}, 0, 1\right), \qquad d_{\text{cap}} > 0. \tag{32}$$

The fixed d_{cap} is calibrated from training-side initialization rollouts. In expierment we use d_cap equal to four. At zero relaxation, the discrepancy reduces exactly to $N^{-1}\sum_i C_{ii}$. The reward is in $[-1,1]$ and is evaluated once per completed video. Clipping can make sufficiently large errors indistinguishable. It is not backpropagated through point construction or nearest-neighbor matching.

This is the only Stage-III alignment reward. Contact annotations may be available from simulation, but contact, momentum, penetration, and material states are not explicit reward terms.

What the reward measures. This reward measures time-indexed dynamic-region geometry with bounded local temporal relaxation. Independent frame matches need not form a one-to-one or globally monotone correspondence, and the point-set distance does not preserve point identity. Predicted flow is used for validity, not cross-frame reward correspondences. Identity-sensitive behavior is evaluated separately with CoTracker3-derived worldlines from generated RGB (Appendix H.4); this diagnostic is not part of the training reward and does not itself guarantee persistent object identity.

D Training Details

This section consolidates the stage-wise training protocol, compute footprint, and inference overhead of Phys4D for reproducibility. Unless otherwise noted, the reported training details correspond to WAN2.2-5B, which is also the backbone used for the main ablation studies.

D.1 Stage-wise Training Protocol

Phys4D follows the same three-stage training pipeline described in the main paper. Stage I trains only the auxiliary depth and motion heads while keeping the pretrained DiT video backbone frozen, thereby constructing the RGB-D-motion interface without altering RGB generation. Stage II then fine-tunes lightweight high-noise LoRA adapters together with the auxiliary heads using simulation-derived geometry, motion, and warp supervision. Stage III keeps the backbone frozen and further optimizes the same adapter space using trajectory-level policy optimization for global physical consistency.

D.2 Training Cost and Inference Overhead

Training cost. All main ablations are conducted on WAN2.2-5B. Stage I trains only the auxiliary depth and motion heads with the frozen DiT backbone and takes approximately 1 day on 8×H100 GPUs. Stage II fine-tunes high-noise LoRA adapters and takes approximately 2 days. Stage III optimizes the same LoRA parameters via trajectory-level policy optimization and takes approximately 1.5 days. The total wall-clock cost is approximately 4.5 days on 8×H100 GPUs, i.e., about 864 H100 GPU-hours. This is a one-time post-training cost of the WAN2.2-5B backbone rather than a recurring deployment cost; Appendix F.5 reports the separate Physics-IQ Verified comparison with larger native backbones.

Inference overhead. Phys4D adds lightweight LoRA adapters and auxiliary depth/motion heads. The auxiliary heads are not part of the iterative denoising loop, so inference remains close to the original backbone. We measure per-video inference time for the base model and the Phys4D-enhanced model. The measured runtimes are summarized in Table 7.

Analysis. Phys4D introduces only small inference overhead: +2s for Open-Sora V1.2 (+5.0%), +5s for WAN2.2-5B (+4.2%), and +3s for CogVideoX-5B (+4.0%). The deployed backbone size

Table 7: **Inference overhead.** Phys4D introduces only slight inference slowdown because it uses lightweight LoRA adapters and small auxiliary heads outside the iterative denoising loop.

Model	Baseline ↓	+ Phys4D ↓	Slowdown
Open-Sora V1.2	40s/video	42s/video	+2s
WAN2.2-5B	120s/video	125s/video	+5s
CogVideoX-5B	75s/video	78s/video	+3s

is unchanged, so the memory footprint at inference is essentially that of the base model plus the LoRA adapters. This supports the design goal of injecting physical structure without substantially increasing inference cost.

D.3 Stage III RL Training Details and Curves

Motivation. Stage II provides dense local supervision through depth, motion, and warp consistency, but these losses are local: they mainly constrain adjacent frames and cannot directly optimize sequence-level physical outcomes such as trajectory drift, penetration, or delayed object-interaction failures. Stage III therefore uses a trajectory-level policy optimization objective to correct residual global physical violations after Stage II has established a locally coherent RGB-D-motion interface.

Rollouts and policy states. The stochastic denoising policy and its state/action definitions are given in Appendix I. Each rollout uses a paired simulator context: the initial state, physical parameters, camera, initial image, prompt, and reference trajectory are shared within a group; generation seeds vary.

Trainable parameters. During Stage III, the pretrained DiT backbone remains frozen. We optimize only the lightweight adapter parameters initialized from Stage II. The depth and motion heads and learned gate are frozen for reward evaluation. This confines simulation-driven updates to a low-rank adapter space and reduces the risk of overfitting to simulator-specific appearance statistics.

Reward. After decoding a complete video, we compute the locally time-relaxed discrepancy in Eq. (31) and map it to the terminal reward $R_{4D} \in [-1, 1]$ using Eq. (32). The temporal radius stays fixed at one frame; λ_t changes only the penalty for an allowed offset. Reward construction and nearest-neighbor selection are not differentiated through the generated video.

Optimization. We use PPO-style optimization with group-normalized terminal rewards. The rollout policy is fixed during a collection round and refreshed before the next; the KL reference remains the Stage-II checkpoint. We normalize rewards within each complete paired context and assign one advantage to every optimized denoising transition of that video. No value network or generalized advantage estimator is used. The probability-ratio surrogate, KL reduction, and advantage definition are specified once in Appendix I.3.

Training cost. Stage III takes approximately 1.5 days on 8×H100 GPUs when initialized from Stage II. Together with Stage I and Stage II, the full training pipeline takes approximately 4.5 days on 8×H100 GPUs for WAN2.2-5B. Stage III does not introduce additional inference-time iterative modules; after training, inference only uses the lightweight adapters and the same denoising process as the base video generator.

Analysis. The reward traces in Fig. 5 show the reported training trend. Stage-wise outcome differences are analyzed in Appendix G.1; the figure does not plot Chamfer or KL.

Stage-III configuration. Tables 8 and 9 specify the Stage-III configuration for WAN2.2-5B. The policy is initialized from a Stage-II checkpoint, and only the LoRA parameters are updated. The pretrained backbone, depth head, motion head, and learned gate remain frozen.

Alignment and reward conventions. Here, D denotes the nonnegative sequence-level discrepancy obtained by bidirectional local frame matching. Each frame-pair cost is the sum of the two directed mean squared 3D nearest-neighbor distances. The search radius remains $r = 1$ for every value of λ_t.

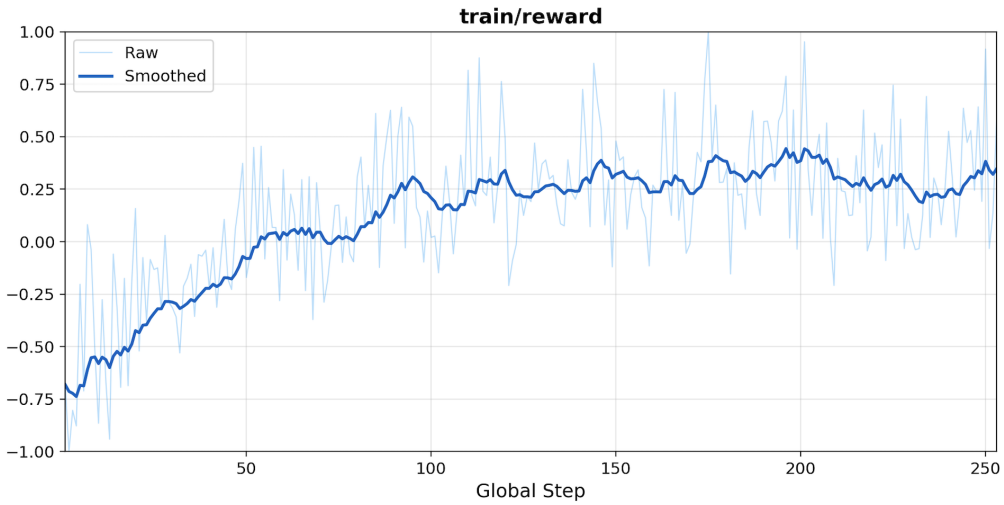

Figure 5: **Stage III RL training curves.** Stage III starts from the Stage II checkpoint and optimizes only lightweight adapter parameters while keeping the pretrained DiT backbone frozen. The plotted raw and smoothed reward traces show the reported training trend.

Table 8: Stage-III configuration: model, alignment reward, and rollout collection. The temporal radius is fixed to one frame; λ_t controls offset tolerance within this fixed neighborhood, not its width.

Hyperparameter	**Setting**
Model and parameter updates	
Backbone	WAN2.2-5B
Initialization	Stage-II checkpoint
Trainable modules	LoRA adapters only
Frozen modules	Backbone, depth head, motion head, and gate
Local temporal alignment	
Matching unit	Frame pair
Frame-pair discrepancy	Symmetric 3D Chamfer: sum of two directed means
Pointwise spatial distance	Squared Euclidean distance
Temporal radius	$r = 1$ frame
Temporal relaxation	$\lambda_t = 0.03$
Allowed reference frames	$\{i-1, i, i+1\}$, restricted to valid frame indices
Temporal offset penalty	$(\tau_i - \tau_j)^2/\lambda_t^2$ for $\lambda_t > 0$
Zero-relaxation behavior	Exact diagonal matching only ($i = j$)
Outer matching	Bidirectional local minimum, averaged over frames and directions
Reward calibration	
Reward mapping	$R = 1 - 2\ \mathrm{clip}(D/d_{\mathrm{cap}}, 0, 1)$
Reward range	$[-1, 1]$
Calibration quantile	0.95
Calibration data	Training split only
Scale after calibration	Fixed
Per-batch min–max normalization	Disabled
Rollout collection	
Contexts per iteration	8
Videos per context	$G = 8$
Total videos per iteration	64
Old-policy refresh	Before each rollout iteration; fixed during its updates

At $\lambda_t = 0$, only same-frame matches are allowed, so D reduces to the mean per-frame symmetric 3D Chamfer distance. For $\lambda_t > 0$, an allowed neighboring-frame match additionally incurs $(\tau_i - \tau_j)^2/\lambda_t^2$; increasing λ_t reduces this penalty without admitting offsets beyond one frame. The timestamps τ_i refer to video time, not denoising time. No per-batch min–max reward normalization is used.

Table 9: Stage-III optimization and pilot budget. Batch sizes count complete videos, not individual denoising transitions. The pilot budget excludes calibration and evaluation samples; update counts assume unchanged sample counts.

Hyperparameter	setting
Optimizer	
Algorithm	PPO-style, group-normalized advantages
Optimizer	AdamW
Learning rate	1×10^{-5}
Adam coefficients	$(\beta_1, \beta_2) = (0.9, 0.999)$
Adam numerical constant	$\epsilon_{\text{Adam}} = 10^{-8}$
Weight decay	10^{-4}
Global minibatch size	16 videos
Inner epochs per rollout batch	1
Maximum gradient norm	1.0
Learning-rate warmup	20 optimizer updates
Schedule after warmup	Constant
PPO surrogate and reference regularization	
PPO clipping parameter	$\epsilon_{\text{PPO}} = 10^{-3}$
Log-probability reduction	Mean over effective latent dimensions
Reference policy	Fixed Stage-II checkpoint
Reference KL coefficient	$\beta_{\text{KL}} = 0.01$
KL reduction	Mean over latent dimensions, then training denoising steps
Denoising-loss reduction	Mean over training denoising steps within each video
Entropy coefficient	0
Advantage estimation	
Normalization group	Same full simulator context
Standard deviation	Population standard deviation
Numerical constant	$\epsilon_A = 10^{-4}$
Advantage clipping	$[-5, 5]$
Discount factor	$\gamma = 1$
Critic / GAE	Neither used
Precision and pilot budget	
Forward-pass precision	BF16
Probability and statistics precision	FP32
Rollout iterations	50
Training videos	3,200
Expected optimizer updates	200

Grouping and reduction conventions. A group contains eight videos generated under the same complete simulator context, rather than merely the same prompt. For rewards $R_1, \ldots, R_G$, the group statistics and advantages are computed as

$$\begin{aligned} \bar{R}_s &= \frac{1}{G}\sum_{g=1}^{G} R_g, & \sigma_s^2 &= \frac{1}{G}\sum_{g=1}^{G}(R_g - \bar{R}_s)^2, \\ \widehat{A}_g &= \text{clip}\left(\frac{R_g - \bar{R}_s}{\sigma_s + \epsilon_A}, -5, 5\right), \quad G = 8, & \epsilon_A &= 10^{-4}. \end{aligned} \tag{33}$$

The same terminal advantage is assigned to the participating denoising steps of each video. Per-step log-probabilities are averaged over effective latent dimensions before exponentiating the new–old difference to form the clipping ratio. This is a dimension-normalized PPO-style surrogate, not the full-joint likelihood ratio. KL terms are averaged over latent dimensions and then over participating training denoising steps. The rollout policy is refreshed before each collection iteration, whereas the Stage-II reference remains fixed throughout training. With 64 rollout videos, a global minibatch of 16 videos, and one inner epoch, each iteration gives four optimizer updates; the 50-iteration pilot therefore gives 200 updates, provided that the sample counts remain unchanged.

E QUALITATIVE RESULTS

Figure 7 and Figure 6 qualitatively compare Phys4D with the base video diffusion model across a diverse set of physical interaction scenarios under static camera settings, including rigid-body manipulation, multi-object dynamics, soft-body deformation, fluid flow, and contact-rich interactions. While the baseline model often generates visually coherent frames, it frequently violates object-level physical constraints, leading to shape distortion, object duplication, incorrect interaction outcomes, or implausible motion patterns over time.

In manipulation scenarios, such as object placement and grasping, baseline models tend to introduce non-physical deformations or geometry drift, even for nominally rigid objects. Phys4D, in contrast, preserves object integrity throughout the interaction, maintaining consistent geometry and contact behavior as objects are manipulated or released. For multi-object scenes, baseline generations commonly suffer from object count inconsistencies and unstable trajectories, whereas Phys4D maintains correct object identities and produces smooth, coherent motion consistent with the underlying scene layout.

We further observe substantial improvements in fluid-object interactions. In pouring scenarios, baseline models often violate basic physical principles, such as gravity alignment or container geometry, resulting in fluid emerging from incorrect locations. Phys4D generates fluid motion that is directionally consistent with gravity and constrained by the container structure, yielding more realistic flow behavior over time.

Across all scenarios, these qualitative results highlight that Phys4D goes beyond frame-wise visual plausibility, enforcing object-level continuity, interaction consistency, and physically grounded dynamics. This supports the claim that Phys4D captures a coherent observable geometry-motion interface, rather than synthesizing isolated, appearance-consistent frames.

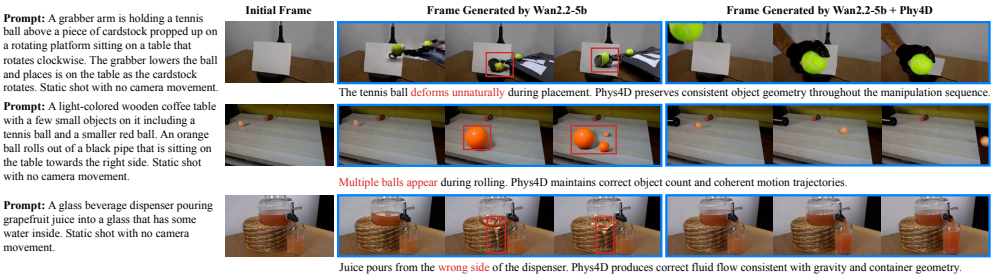

Figure 6: **Qualitative comparison on Physics-IQ scenarios.** We compare WAN2.2-5B baseline (middle) with WAN2.2-5B + Phys4D (right) across three physical interactions. Phys4D improves geometry consistency, motion plausibility, and temporal stability, while the baseline shows shape distortion and incoherent physical behavior.

F EXPERIMENTS

F.1 SHARED EVALUATION SETTINGS AND REPORTING CONVENTIONS

This section provides additional experimental details and full quantitative results. We evaluate whether Phys4D transfers simulation-derived physical structure to generated RGB videos using three external video-level physics evaluations: Physics-IQ (Motamed et al., 2025), VBench-2.0 Physics (Zheng et al., 2025), and PhyGenBench (Meng et al., 2025). Within each benchmark, the base and Phys4D-enhanced models use matched prompts, conditioning inputs, sampling settings, video length, and evaluator. Physics-IQ Original, Physics-IQ Verified full set, and the Verified AC35 subset remain separate protocols and are never compared as if they shared a score scale. Where reported, main-text uncertainty denotes one standard deviation over five generation seeds for a fixed checkpoint, not a confidence interval.

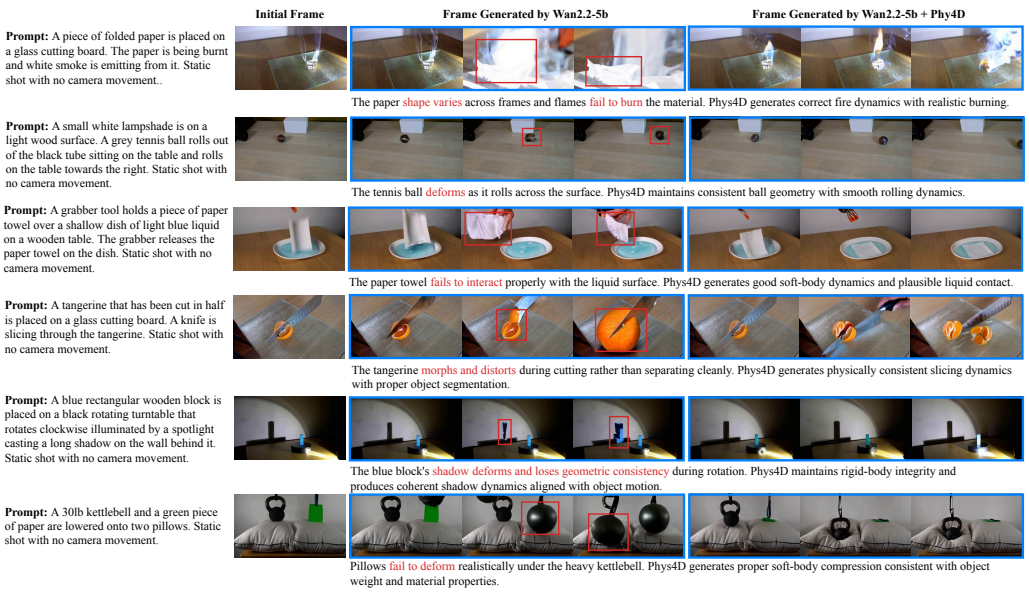

Figure 7: **Qualitative Result of a Phys4D-Generated Scene.**

F.2 PHYSICS-IQ ORIGINAL

Benchmark and protocol. Physics-IQ (Motamed et al., 2025) evaluates whether video generation models can predict physically plausible future videos from real-world captured physical interactions. Each test case provides a switch frame and a text description, and the model generates a 5-second continuation. Following the official protocol, generated videos are compared with ground-truth future videos using four complementary metrics: MSE, Spatiotemporal IoU, Spatial IoU, and Weighted Spatial IoU. MSE measures pixel-level future-frame reconstruction error, while the IoU metrics evaluate whether the generated motion occurs at the correct location, time, and magnitude. The final Physics-IQ score aggregates these metrics into a normalized physical prediction score.

For evaluation, we generate 5-second videos conditioned on the switch frame following the Physics-IQ protocol. We use the same stochastic sampling setup for the base and Phys4D-enhanced models, with temperature 1.0 and top-$p = 0.95$ where applicable. We report matched-protocol comparisons only. Public leaderboard numbers obtained under different model checkpoints, input modalities, preprocessing, or generation settings are not mixed with our same-backbone comparisons. The full matched-protocol results are presented in Table 10.

Benchmark version: Physics-IQ Original, frozen January 2026 snapshot. Unless stated otherwise, all Physics-IQ numbers in the main paper and in this appendix are computed with the *original* Physics-IQ workflow (Motamed et al., 2025), frozen at a January 2026 snapshot of the official repository: official switch frames, original prompts, five-second continuations, ground-truth future videos and masks, and the official scoring code. "Matched switch-frame I2V protocol" therefore does not denote a custom protocol. It means that, for each backbone, the base model and its Phys4D-enhanced counterpart are generated by us under identical conditioning inputs, prompts, FPS, resolution, duration, sampling settings, preprocessing, and evaluator.

The separate *Physics-IQ Verified* release (Rädsch et al., 2026) uses clarified prompts, corrected ground-truth masks, and equally weighted sample-level scoring in place of dataset-level aggregation. We use Physics-IQ Verified July Version to froze Physics-IQ Verified Benchmark. These benchmark-version differences materially change absolute scores and model rankings, so Original and Verified scores must not be compared directly. Two consequences matter for reading our tables. First, our WAN2.2-5B base score of 16.8 and the public leaderboard entry of about 32.2 for "Wan2.2" are neither the same checkpoint nor the same benchmark version: the public entry is the WAN2.2-I2V-A14B checkpoint evaluated under Verified. In our controlled evaluation, switching the *same* 5B checkpoint from Original to Verified changes its score by only $+4.5$ ($16.8 \rightarrow 21.33$), whereas under the identical Verified protocol the 5B base (21.33) and the public A14B entry (32.2) still differ by about 10.9

points (Rädsch et al., 2026). The discrepancy is thus explained by checkpoint scale and inference settings, not by generation length, resolution, or the switch-frame setting. Second, several third-party leaderboard entries cannot be regenerated by us because their executable generation code and exact inference configurations are not public; we list such entries only in the reference-only Table 11 and never mix them with the controlled comparisons. Section F.5 reports our own checkpoints evaluated under the Verified protocol.

Table 10: **Full Physics-IQ results under the matched switch-frame I2V protocol (Physics-IQ Original, frozen January 2026 snapshot).** MSE measures future-frame reconstruction error. ST-IoU, S-IoU, and WS-IoU measure spatiotemporal, spatial, and weighted spatial motion-mask overlap, respectively. Phys4D consistently improves all Physics-IQ metrics across multiple backbones.

Backbone	Params	Method	MSE ↓	ST-IoU ↑	S-IoU ↑	WS-IoU ↑	Score ↑
CogVideoX	5B	Base	0.013	0.116	0.222	0.142	18.8
CogVideoX	5B	+ Phys4D	**0.008**	**0.182**	**0.270**	**0.169**	**32.8**
WAN2.2	5B	Base	0.016	0.088	0.150	0.105	16.8
WAN2.2	5B	+ Phys4D	**0.013**	**0.158**	**0.238**	**0.149**	**30.9**
Open-Sora V1.2	1.1B	Base	0.021	0.072	0.135	0.092	14.5
Open-Sora V1.2	1.1B	+ Phys4D	**0.011**	**0.119**	**0.226**	**0.158**	**24.7**

Analysis. Phys4D achieves substantial improvements over base video diffusion models across multiple backbones. When applied to CogVideoX-5B, Phys4D improves Spatiotemporal IoU from 0.116 to 0.182, a 56.9% relative gain, and Spatial IoU from 0.222 to 0.270, a 21.6% relative gain, while reducing MSE from 0.013 to 0.008, a 38.5% reduction. Similarly, Phys4D improves WAN2.2-5B's Spatiotemporal IoU from 0.088 to 0.158, a 79.5% relative gain, and Open-Sora V1.2 from 0.072 to 0.119, a 65.3% relative gain. The overall Physics-IQ score also improves from 18.8 to 32.8 on CogVideoX-5B, from 16.8 to 30.9 on WAN2.2-5B, and from 14.5 to 24.7 on Open-Sora V1.2. These consistent improvements across architectures indicate that the physics-grounded training paradigm is architecture-agnostic and effectively transfers to different video diffusion backbones.

Notably, Phys4D on CogVideoX-5B achieves the best Physics-IQ score among the reported same-backbone comparisons, with an MSE of 0.008, Spatiotemporal IoU of 0.182, Spatial IoU of 0.270, Weighted Spatial IoU of 0.169, and overall score of 32.8. This suggests that simulation-grounded structure injection can substantially improve the physical prediction ability of open-source video diffusion models.

Uncertainty, seeds, and sample counts. All benchmark scores are averages over the full official test set of the respective benchmark rather than over selected examples. The $\pm$ values in Table 1 denote the mean $\pm$ one standard deviation over five independent generation seeds for the same checkpoint and prompts; they are not confidence intervals. The per-metric and per-dimension breakdown tables in this appendix (Tables 10, 12, and 13) report the corresponding means without uncertainty bars. The observed gains are consistent across backbones, metrics, benchmark versions (Sec. F.5), and ablations, providing convergent evidence that Phys4D improves observable physical plausibility.

Original-protocol reference comparison. For broader context, Table 11 lists public Physics-IQ Original reference scores. Different model checkpoints and input interfaces make these rows descriptive rather than a matched comparison. In particular, the WAN2.2-5B + PhysGen3D value of 20.4 belongs to this Original reference and is not its Verified AC35 result.

F.3 VBench-2.0 Physics

Benchmark and protocol. VBench-2.0 (Zheng et al., 2025) extends video generation evaluation from superficial faithfulness, such as visual quality and temporal smoothness, to intrinsic faithfulness. Its Physics dimension evaluates whether generated videos follow real-world physical principles across four sub-dimensions: Mechanics, Thermotics, Material, and Multi-View Consistency. These sub-dimensions assess physical state changes, thermal transformations, material behavior, and geometry consistency under viewpoint or camera changes.

Table 11: **Physics-IQ Original reference comparison.** Protocols and model interfaces may differ; the matched same-backbone results in Table 10 provide the controlled comparison.

Model	Model Size	Original Score ↑
WAN2.2-5B + Phys4D	5B	**30.9**
Veo3	N/A	23.2
Runway Gen 3	N/A	22.8
WAN2.2-5B + PhysGen3D	5B	20.4
VideoPoet	8B	20.3
Lumiere	4B	19.0
WAN2.2-5B	5B	16.8
Stable Video Diffusion	1.1B	14.8
Pika 1.0	N/A	13.0
Sora	N/A	10.0

Table 12: **Full VBench-2.0 Physics results.** VBench-2.0 Physics evaluates intrinsic physical faithfulness across Mechanics, Thermotics, Material, and Multi-View Consistency. Phys4D improves the overall Physics Score across all three backbones and improves all individual sub-dimensions.

Backbone	Params	Method	Mechanics ↑	Thermotics ↑	Material ↑	Multi-View Consistency ↑	Physics Score ↑
CogVideoX	5B	Base	0.5982	0.5618	0.4572	0.3224	0.4849
CogVideoX	5B	+ Phys4D	**0.7284**	**0.6706**	**0.5002**	**0.4824**	**0.5954**
WAN2.2	5B	Base	0.6220	0.5786	0.4255	0.3450	0.4928
WAN2.2	5B	+ Phys4D	**0.7604**	**0.7002**	**0.5418**	**0.4964**	**0.6247**
Open-Sora V1.2	1.1B	Base	0.5821	0.5508	0.4083	0.3079	0.4623
Open-Sora V1.2	1.1B	+ Phys4D	**0.7147**	**0.6603**	**0.5068**	**0.4778**	**0.5899**

We follow the official VBench-2.0 Physics prompt suite and evaluation pipeline. For each backbone, the base model and the Phys4D-enhanced model generate videos using the same prompts and sampling settings. We report each physics sub-dimension and the overall Physics Score, computed as the average over the four sub-dimensions. Higher scores indicate stronger intrinsic physical faithfulness. Table 12 provides the complete per-backbone breakdown.

Analysis. Phys4D consistently improves the overall VBench-2.0 Physics Score across all three backbones. CogVideoX-5B improves from 0.4849 to 0.5954, a 22.8% relative gain; WAN2.2-5B improves from 0.4928 to 0.6247, a 26.8% relative gain; and Open-Sora V1.2 improves from 0.4623 to 0.5899, a 27.6% relative gain. The improvements are especially clear in Mechanics, Thermotics, and Multi-View Consistency, indicating that Phys4D improves both physical state evolution and geometry consistency. For example, WAN2.2-5B improves from 0.6220 to 0.7604 in Mechanics and from 0.3450 to 0.4964 in Multi-View Consistency. In this matched evaluation, Phys4D also improves the Material sub-dimension across all three backbones.

F.4 PhyGenBench

Benchmark and protocol. PhyGenBench (Meng et al., 2025) evaluates physical commonsense in text-to-video generation. It contains prompts covering four physical domains: Mechanics, Optics, Thermal phenomena, and Material properties. The benchmark is designed around explicit physical laws and observable physical phenomena, such as gravity, buoyancy, reflection, refraction, melting, boiling, hardness, and solubility. Its evaluator, PhyGenEval, uses a hierarchical protocol to assess whether the generated video contains the expected physical phenomenon, follows the correct physical order, and remains natural over the full video.

We use the official PhyGenBench prompt suite and PhyGenEval pipeline. For each prompt, we generate videos with the base backbone and the corresponding Phys4D-enhanced model under the same sampling settings. We report scores for Mechanics, Optics, Thermal, and Material categories, as well as their average. Higher scores indicate stronger physical commonsense correctness. The full per-category results appear in Table 13.

Analysis. Phys4D improves PhyGenBench scores across all three backbones. CogVideoX-5B improves from 0.45 to 0.58, a 28.9% relative gain; WAN2.2-5B improves from 0.45 to 0.57, a

Table 13: **Full PhyGenBench results.** PhyGenBench evaluates physical commonsense generation across Mechanics, Optics, Thermal phenomena, and Material properties. Phys4D consistently improves the average score across all three backbones and improves all per-category scores.

Backbone	Params	Method	Mechanics ↑	Optics ↑	Thermal ↑	Material ↑	Average ↑
CogVideoX	5B	Base	0.40	0.56	0.41	0.43	0.45
CogVideoX	5B	+ Phys4D	**0.53**	**0.66**	**0.57**	**0.56**	**0.58**
WAN2.2	5B	Base	0.42	0.50	0.42	0.46	0.45
WAN2.2	5B	+ Phys4D	**0.52**	**0.63**	**0.56**	**0.57**	**0.57**
Open-Sora V1.2	1.1B	Base	0.43	0.50	0.44	0.39	0.44
Open-Sora V1.2	1.1B	+ Phys4D	**0.53**	**0.62**	**0.57**	**0.56**	**0.57**

Table 14: **Physics-IQ Verified.(July Version)** Rows marked *Ours* are generated and scored by us under the official Verified protocol. Rows marked *Official* are quoted from the Verified leaderboard (Rädsch et al., 2026) and serve as a cross-scale reference rather than a controlled comparison. Relative model size is given with respect to the 5B backbones.

Source	Model	Parameters	Training variant	Physics-IQ Verified ↑	Gain over same base
Ours	WAN2.2-5B	5B (1.0×)	Base	21.33	–
Ours	WAN2.2-5B	5B (1.0×)	+ Stage II	32.20	+10.87
Ours	WAN2.2-5B	5B (1.0×)	+ Full Phys4D	**35.80**	**+14.47**
Ours	CogVideoX-5B	5B (1.0×)	Base	25.38	–
Ours	CogVideoX-5B	5B (1.0×)	+ Full Phys4D	**37.20**	**+11.82**
Official	WAN2.2-I2V-A14B (Wan et al., 2025)	14B active / 27B total	Native model	32.2 ± 0.6	–
Official	Cosmos3-Nano (NVIDIA, 2026)	16B (3.2×)	Native model	30.3 ± 0.6	–
Official	Cosmos3-Super (NVIDIA, 2026)	64B (12.8×)	Native model	$\mathbf{39.5 \pm 0.8}$	–

26.7% relative gain; and Open-Sora V1.2 improves from 0.44 to 0.57, a 29.5% relative gain. The improvements are consistent across Mechanics, Optics, Thermal, and Material categories. For instance, CogVideoX-5B improves from 0.40 to 0.53 in Mechanics and from 0.41 to 0.57 in Thermal, while WAN2.2-5B improves from 0.50 to 0.63 in Optics and from 0.46 to 0.57 in Material. These results suggest that Phys4D improves not only low-level motion prediction but also semantic physical commonsense as judged by an independent VLM-based evaluation pipeline.

F.5 Physics-IQ Verified Evaluation and Cross-Scale Comparison

Motivation. Table 10 uses the frozen Original snapshot so that all ablations in this paper remain mutually comparable. To remove benchmark-version ambiguity and compare against larger image-to-video models under one protocol, we also evaluate our checkpoints under the official *Physics-IQ Verified* release (Rädsch et al., 2026)(Frozen at July), using its data, corrected masks, best-practice prompts, generation protocol, and scoring pipeline. All results below use the I2V setting. Official entries are quoted from the Verified leaderboard, since their code and inference configurations are not available to us for controlled evaluations.

Analysis. The controlled within-backbone rows in Table 14 provide the primary evidence and show the same within-backbone pattern under the Verified benchmark. On WAN2.2-5B, Stage II improves the Verified score by 10.87 points and Stage III contributes a further 3.60 points, for a total gain of 14.47 points without increasing backbone size; the same recipe improves CogVideoX-5B by 11.82 points. As a cross-scale reference, both Phys4D-enhanced 5B models exceed the official WAN2.2-I2V-A14B (14B active) and Cosmos3-Nano (16B) entries, and CogVideoX-5B + Phys4D is 2.3 points below the 64B Cosmos3-Super. We do not claim that a Phys4D-enhanced 5B model necessarily outperforms every larger native model, nor that the Original-protocol score of 30.9 is directly comparable to the evolving public leaderboard. The claim supported by both benchmark versions is the controlled one: for the same pretrained backbone under the same evaluation protocol, Phys4D substantially improves physical consistency while preserving the backbone's real-video generative prior.

Headroom on stronger backbones. Phys4D is a model-agnostic post-training framework rather than a substitute for model scaling. At the time of our experiments, the most relevant stronger open I2V backbone was MAGI-1 24B (Sand.ai, 2025), whose full three-stage adaptation exceeded our compute budget, and the Cosmos 3 family (NVIDIA, 2026) was released after our experiments were

Table 15: **Video-level physics on the unified Physics-IQ Verified AC35 protocol.** All methods use the same 35 scenarios and 105 take-1 videos, initial observation, event information, and scoring pipeline, with method-native input encoding. These are subset scores, not full-set Verified or Original scores.

Method	Verified AC35 ↑
WAN2.2-5B Base	19.62
PhysGen3D	21.4
TesserAct	22.6
RealWonder	23.40
WAN2.2-5B + Phys4D	**32.68**

Table 16: **Representative 4D diagnostics on held-out simulated scenes.** Chamfer is set-level geometry. Mean Drift and Fail Rate use the shared CoTracker3 worldline protocol on generated RGB, with row-specific depth and simulator trajectory references. Complete rows are in Table 23.

Method	4D Chamfer ↓	Mean Drift ↓	Fail Rate ↓
PhysGen3D	0.4718	0.4962	10.24%
TesserAct	0.4596	0.4819	9.83%
Phys4D + WAN2.2 native	**0.4159**	**0.4312**	**8.36%**

complete. Whether the gain diminishes as the base model becomes stronger therefore remains open; the Verified results above indicate that structural physics injection is large enough to close part of the model-scale gap for 5B backbones.

One-time post-training cost versus deployment cost. The complete cost and measured inference overhead are reported in Appendix D.2 and Table 7. The WAN2.2-5B post-training cost is approximately 864 H100 GPU-hours once, whereas its measured inference time rises from 120 to 125 seconds per video. External models do not have comparable reported training and inference configurations, so their Verified scores above are cross-scale references rather than a controlled cost comparison.

F.6 Comparison with 3D/4D and Physics-Aware Generation Baselines

Comparison setting. We compare Phys4D with RealWonder (Liu et al., 2026), TesserAct (Zhen et al., 2025), and PhysGen3D (Chen et al., 2025a) using the same initial observation, event description, and test-case selection. Each method receives this information through its native interface; shared information does not imply identical input tensors. Where a method needs an explicit control, it is constructed from the event description using fixed rules, without the ground-truth future or per-case outcome tuning. RealWonder's prompt-derived actions use globally fixed magnitudes and durations per action family. We report video-level physics and held-out simulation diagnostics in separate panels because they have different data and metrics.

Video-level physics on Verified AC35. All five methods use the same unified Physics-IQ Verified evaluator on the Action-Compatible subset of 35 scenarios and 105 take-1 videos. This subset is selected from descriptions, includes supported solid, fluid, and deformable cases, and retains failures. It is distinct from the full Verified set in Appendix F.5 and Original in Appendix F.2. The unified AC35 scores are reported in Table 15.

Geometry–motion diagnostics on held-out simulation. Table 16 shows representative values from the full seven-row 4D table in Appendix H.4. These diagnostics are not Physics-IQ scores.

Analysis. On Verified AC35, Phys4D scores 32.68 versus 19.62 for its base, 23.40 for RealWonder, 22.6 for TesserAct, and 21.4 for PhysGen3D. On the separate held-out simulation diagnostics, its 4D Chamfer, RGB-tracked mean drift, and failure rate are lower than those of the two structured baselines. These compare complete pipelines under different native interfaces and separate evaluation domains, limiting causal interpretation; tracker errors remain part of the worldline measurements.

F.7 Long-Video Generation with Self-Forcing

Setting. We further evaluate whether Phys4D remains compatible with longer-video generation using a self-forcing strategy on VBench2.0-Long. This is not a physics-specific benchmark, so we do not use it as primary evidence for physical consistency. Instead, it provides an auxiliary check that Phys4D does not degrade longer-video quality or semantic consistency.

Analysis. As shown in Table 17, Phys4D improves the total score from 82.44 to 85.66, the quality score from 84.58 to 87.72, and the semantic score from 73.8 to 77.4. These results suggest that Phys4D remains compatible with longer-video generation pipelines and does not harm general long-video quality or semantic consistency. We nevertheless treat long-horizon physical simulation as beyond the scope of this work, since current backbones are still primarily short-video generation models and VBench2.0-Long is not physics-specific.

Table 17: **VBench2.0-Long with self-forcing.** Phys4D improves the total score, quality score, and semantic score under a longer-video self-forcing setting.

Model	Total Score ↑	Quality Score ↑	Semantic Score ↑
WAN2.2 + Self-Forcing	82.44	84.58	73.8
WAN2.2 + Phys4D + Self-Forcing	**85.66**	**87.72**	**77.4**

F.8 Visual Prior Preservation

Motivation. RQ3 evaluates whether injecting simulation-derived physical structure into pretrained video generators harms their real-video visual prior. This is important because Phys4D is designed to transfer physical structure from simulation, not simulator-specific appearance. A failure mode of simulation-based training would be that the model improves physical metrics while degrading open-domain visual quality, color fidelity, style adherence, or human motion generation. Therefore, RQ3 focuses on whether the Phys4D-enhanced models remain visually close to their corresponding base models.

Benchmark choice. We use VBench (Huang et al., 2023) for this evaluation because it provides disentangled measurements of video generation quality across visual and semantic dimensions. Unlike the physics-focused diagnostics used in RQ1, VBench is better suited for measuring whether the pretrained real-video prior is preserved after Phys4D training. We therefore use VBench as a visual-prior preservation benchmark rather than as a physics benchmark.

Evaluated dimensions. We report five VBench dimensions in Table 3: *Aesthetic Quality*, *Imaging Quality*, *Color*, *Appearance Style*, and *Human Action*. The first three dimensions directly measure low-level and perceptual visual quality. *Aesthetic Quality* reflects the overall visual attractiveness of generated videos, *Imaging Quality* measures image-level fidelity and degradation artifacts, and *Color* evaluates color consistency and naturalness. We define **Visual Avg.** as the average of these three appearance-oriented dimensions:

$$\text{Visual Avg.} = \frac{\text{Aesthetic Quality} + \text{Imaging Quality} + \text{Color}}{3}. \tag{34}$$

In addition, we report *Appearance Style* and *Human Action*. These two dimensions are not included in Visual Avg.; instead, they are used to check whether Phys4D affects style-level prompt adherence and open-domain human motion generation.

Evaluation protocol. For each backbone, we compare the original base model with its Phys4D-enhanced counterpart under the same VBench prompt set, sampling configuration, output duration, resolution, and evaluation pipeline. No VBench-specific tuning is applied to the Phys4D models. This matched protocol ensures that differences in Table 3 reflect the effect of Phys4D rather than changes in generation settings or evaluation conditions. The purpose of this experiment is not to maximize VBench visual scores, but to verify that Phys4D does not substantially damage the pretrained real-video visual prior while improving physical plausibility in RQ1 and RQ2.

Table 18: **Reported VBench-2.0 non-physics prior control.** VB2 Prior Avg. covers Human Fidelity, Creativity, Controllability, and Commonsense. FVD uses the available validation setting and is not used for a domain-specific claim.

WAN2.2-5B variant	VB2 Prior Avg. ↑	FVD ↓
Base	0.586	140.0
Direct simulator-RGB LoRA	0.528	168.5
Full Phys4D (Stage II+III)	0.581	121.5

Implementation details. We evaluate CogVideoX-5B, WAN2.2-5B, and Open-Sora V1.2, together with their Phys4D-enhanced variants. For each model pair, videos are generated using the same prompts and decoding settings. The generated videos are then passed to the official VBench evaluation pipeline to obtain per-dimension scores. For base models, we align the reported scores with the corresponding VBench evaluation setting when available. For Phys4D-enhanced models, we use the same evaluation procedure as the base models to ensure direct comparability.

Results. Table 3 shows that Phys4D largely preserves the visual prior of all three pretrained backbones. The Visual Avg. remains nearly unchanged after Phys4D: CogVideoX changes from 0.7358 to 0.7326, Open-Sora V1.2 changes from 0.7168 to 0.7173, and WAN2.2 changes from 0.7521 to 0.7519. These small changes indicate that Phys4D retains the measured visual quality on these prompts; they do not directly test invariance to simulator rendering.

The individual visual dimensions show the same pattern. For CogVideoX, Aesthetic Quality, Imaging Quality, and Color decrease only slightly after Phys4D. For Open-Sora V1.2, Aesthetic Quality and Color decrease slightly, while Imaging Quality improves. For WAN2.2, Aesthetic Quality improves after Phys4D, while Imaging Quality and Color decrease only slightly. Overall, these results show that the appearance-level behavior of the base video generators is mostly retained.

Beyond the three dimensions included in Visual Avg., Phys4D also maintains or improves *Appearance Style* and *Human Action*. Appearance Style improves from 0.2107 to 0.2184 on CogVideoX, from 0.1972 to 0.2011 on Open-Sora V1.2, and from 0.2139 to 0.2162 on WAN2.2. Human Action also improves from 0.7452 to 0.7728 on CogVideoX, from 0.6872 to 0.7023 on Open-Sora V1.2, and from 0.7433 to 0.7712 on WAN2.2. This suggests that Phys4D does not trade off open-domain action generation for physical structure; instead, the improved physical consistency may also support more coherent motion-related generation.

VBench-2.0 prior control. Table 18 reports the available non-physics VBench-2.0 prior average for the base WAN2.2-5B model, direct simulator-RGB LoRA, and full Phys4D. This is distinct from the VBench 1.0 Visual Avg. in Table 3

Discussion. The RQ3 results complement the physics evaluations in RQ1 and RQ2. While RQ1 shows that Phys4D improves external video-level physics diagnostics and RQ2 shows improvements in the RGB-D-motion world interface, RQ3 verifies that these gains do not come at the cost of degraded visual quality. The nearly unchanged Visual Avg. across all three backbones indicates that the pretrained real-video appearance prior is preserved. At the same time, the consistent improvements in Appearance Style and Human Action suggest that Phys4D remains compatible with open-domain generation behavior. Together with the RGB-FM control in Sec. 6.4, these findings support physical-structure transfer while retaining the measured real-video visual prior; they do not rule out all renderer-specific effects.

G Additional Ablations for RQ4

The main-text RQ4 tests whether most of the observed physics gain persists without simulator-video RGB fitting. This appendix collects the supporting stage, loss, head, and temporal analyses. Unless stated otherwise, values are Physics-IQ Original scores for WAN2.2-5B under the same switch-frame I2V evaluation. Stage I alone is not a separate RGB-generation variant because its backbone is frozen.

Table 19: **Unified ablation and control results on WAN2.2-5B.** All values use Physics-IQ Original. S2 is the full Stage-II reference for head deletions; S3 includes Stage III and is the reference for loss deletions. R0 and H1 happen to score 24.2 but are different training configurations. The intervention column records the reported change, not an additive causal contribution.

ID	Training variant	Reported intervention	Stage III	Physics-IQ ↑	Compare
S0	Base WAN2.2-5B	No Phys4D training	No	16.8	–
S1	Stage III only	Skip Stage II	Yes	18.4	S0
S2	Full Stage II	Local structure training	No	27.8	S0
S3	Full Phys4D	Stage II followed by III	Yes	**30.9**	S2
R0	Direct simulator-RGB LoRA	All-noise RGB adaptation; no Phys4D heads or gate	No	24.2	S0
R1	Stage II w/o simulator-video RGB-FM	Retain structural heads, losses, and gate	No	25.7	S2
H0	Stage II w/o depth and flow heads	Remove both heads and dependent objectives	No	21.4	S2
H1	Stage II with depth head only	Remove motion head and dependent objectives	No	24.2	S2
H2	Stage II with flow head only	Remove depth head and dependent objectives	No	24.8	S2
L0	Full Phys4D w/o geometry warp	Retain heads; remove geometry-warp loss	Yes	25.9	S3
L1	Full Phys4D w/o depth loss	Retain heads; remove depth regression	Yes	23.8	S3
L2	Full Phys4D w/o motion loss	Retain heads; remove motion regression	Yes	24.6	S3
L3	Full Phys4D w/o generated-RGB warp	Retain heads; remove RGB self-consistency	Yes	29.4	S3

G.1 Stage-Wise Ablation

Rows S0–S3 in Table 19 show the reported stage comparison. Stage II changes the Physics-IQ score from 16.8 to 27.8; the Stage-II+III result is 30.9. Direct Stage-III-only training scores 18.4. This pattern supports using local structure training before sequence-level policy optimization. The initialization and evaluator calibration of the Stage-III-only run must be distinguished from the full Stage-III run when interpreting their difference.

G.2 Loss Component Ablation

Rows L0–L3 remove individual objectives while retaining the depth and motion heads; they are loss deletions, not head deletions. Relative to the reported full-method score of 30.9, the scores are 25.9 without geometry warp, 23.8 without depth regression, 24.6 without motion regression, and 29.4 without generated-frame RGB self-consistency. These comparisons support complementary roles for geometry, motion, and temporal coupling in the reported runs.

G.3 Auxiliary Depth and Flow Ablation

Rows H0–H2 compare head configurations against full Stage II (S2=27.8). Removing both heads and their dependent objectives gives 21.4; retaining only the depth head gives 24.2, and retaining only the flow head gives 24.8. These are video-level Physics-IQ scores, so an incomplete RGB-D-motion interface can still be evaluated. H0 retains the Stage-II gate and high-noise RGB-FM; it is not the direct simulator-RGB LoRA control R0. Geometry warp uses simulator flow and therefore does not intrinsically require the predicted flow head, whereas it does require predicted depth. The active objectives of each head-deletion run should follow its recorded implementation.

G.4 Local Temporal Tolerance

The Stage-III discrepancy in Appendix C.6 uses a fixed one-frame radius. Zero relaxation requires same-frame 3D Chamfer; positive λ_t lowers the cost of an allowed neighbor without enlarging the

Table 20: **Stage-III temporal-parameter analysis.** The six variants are evaluated with the shared CoTracker3 worldline protocol. Failure rates are percentages; Length is the evaluator's trajectory-length score. The fixed one-frame matching radius is common to all variants.

λ_t	Physics-IQ ↑	Worldline L2 ↓	Mean Drift ↓	Final Drift ↓	Fail Rate ↓	Length ↑
0	28.8	0.4826	0.4764	0.5018	10.42%	86.01
0.01	29.5	0.4625	0.4568	0.4803	9.55%	87.45
0.02	30.2	0.4481	0.4417	0.4662	8.89%	88.63
0.03	**30.9**	**0.4375**	**0.4312**	**0.4556**	**8.36%**	**89.68**
0.05	29.7	0.4524	0.4469	0.4718	9.18%	88.04
0.08	27.9	0.5017	0.4924	0.5260	11.76%	85.02

window. Table 20 compares six Stage-III variants indexed by their temporal parameter. The fixed neighborhood does not vary across these runs, and the table is a descriptive sensitivity analysis rather than a causal isolation of the matching radius. All variants' worldline values use the shared CoTracker3 protocol in Appendix H.4, with the same query and visibility rules. The depth readout used for these ablation rows should be documented from the evaluation configuration; this does not change the reported scores.

Among the reported rows, the 0.03 variant has the highest Physics-IQ score and lowest RGB-tracked mean drift. This numerical ranking does not by itself isolate a causal effect of the temporal penalty from other optimization interactions.

G.5 Implementation Details of the Simulator-RGB Control

The direct simulator-RGB LoRA control R0 adapts simulator-video latents at all noise levels without Phys4D heads, structural objectives, or gate. The R1 control removes simulator-video RGB flow matching from the gated Stage-II configuration while retaining depth, motion, geometry-warp, generated-frame RGB self-consistency, and the simulator-rendered initial image used as the I2V condition. Its structural branch uses a high-noise conditional forward pass from Gaussian noise, the initial image, and the prompt; simulator-video latents do not supply the RGB-FM target in R1. The full S2 configuration additionally uses high-noise flow matching on latents encoded from simulator videos. All three configurations keep the pretrained backbone frozen and update only their enabled lightweight components.

H Additional 4D World-Level Diagnostics

This section expands RQ2 with controlled diagnostics on held-out simulated scenes. We distinguish structural measurements obtained from generated RGB with shared external estimators from measurements of Phys4D's native RGB-D-motion interface. Simulator annotations provide reference depth, motion, and trajectories; this is not reference-free evaluation. External video-level physics benchmarks are reported separately in RQ1.

We emphasize that this section is not presented as a new benchmark contribution. Instead, it is a diagnostic evaluation designed to answer whether the physical knowledge injected by Phys4D is reflected in geometry, motion, and trajectory-level scene evolution. Throughout this section, "4D" refers to temporally coupled 3D scene evolution represented through RGB-D and motion fields, rather than a dense volumetric 4D reconstruction.

The diagnostics are organized into three levels: (i) per-frame 3D geometry, (ii) local temporal geometry-motion consistency, and (iii) sequence-level 4D world evolution and novel-time continuity. Together, these diagnostics probe whether Phys4D improves the structure exposed by the RGB-D-motion interface rather than only improving RGB-level physics scores.

H.1 Shared Evaluation Setting

All evaluations are conducted on held-out simulated scenes with fixed camera intrinsics and viewpoints. Each method is conditioned on the same initial frame and prompt, and produces a generated video under the same generation settings. The generated video is then lifted into geometry, motion, or

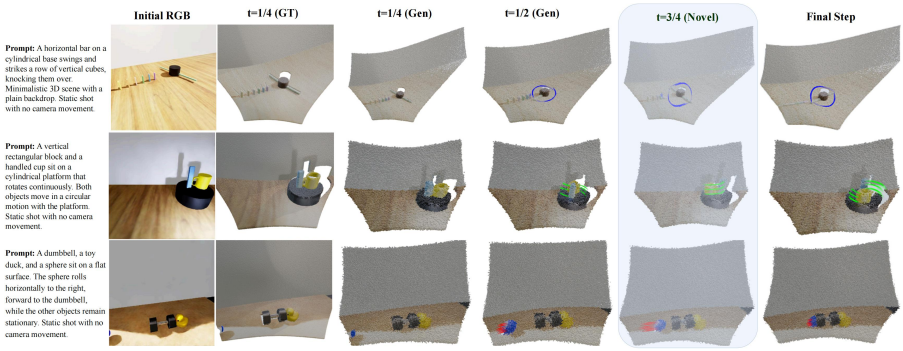

Figure 8: **Qualitative visualization of 4D interface-level diagnostics.** From left to right: ground-truth 4D point cloud; generated 4D point cloud at $1/4$ of the sequence; generated 4D point cloud at $1/2$ of the sequence; novel-time generated 4D point cloud at $3/4$ of the sequence; and generated 4D point cloud at the final frame. Generated points use the Phys4D native-depth interface. These are set-level point clouds, not RGB-tracked object identities or worldlines.

trajectory representations depending on the diagnostic. The scenes come from the 20 templates held out before rollout generation (Appendix A.1); no derived environment or rollout is used in Stage II or Stage III training. Assets, solvers, and physical-parameter ranges may overlap with training, so the split tests unseen templates within supported regimes, not out-of-distribution physics. Five camera videos from one environment observe the same underlying physical rollout, not five independent physical trials.

For RGB-only video generators, we use an off-the-shelf post-hoc lifting pipeline. Specifically, **OTF** denotes off-the-shelf RGB-to-geometry-and-motion lifting, instantiated with DepthAnythingV2 (Yang et al., 2024a) for monocular depth estimation and SEA-RAFT (Wang et al., 2024) for optical flow. For the *Phys4D + WAN2.2 RGB+OTF* row, we apply the same frozen OTF readout to Phys4D-generated RGB. Its matched comparison is WAN2.2 Base + OTF: inputs and generation settings are matched within that backbone, while the external readout is fixed. The *Phys4D + WAN2.2 native interface* row uses the learned Phys4D depth/flow heads. This second contrast is a readout comparison, not another independent RGB-generation gain.

The two Phys4D worldline readouts use identical generated RGB samples and 2D tracks. The same external-depth checkpoint and scale conversion are applied across OTF rows. GT geometry and trajectories serve as scoring references; any reference-assisted depth-scale alignment must be disclosed separately. For worldlines, the reported protocol uses the same frozen CoTracker3 (Karaev et al., 2024) on each method's generated video; the two Phys4D readouts share 2D tracks and differ only in lifting depth. Simulator future-frame flow is restricted to reference construction. The scores below follow the tracker protocol specified in Sec. H.4.

H.2 Per-Frame 3D Geometry

Setting. This diagnostic evaluates whether each generated frame contains accurate instantaneous 3D geometry. We compare predicted depth maps against simulation ground truth under fixed camera intrinsics. The evaluation is frame-wise and does not enforce temporal consistency, so it isolates static geometric accuracy from motion and sequence-level effects.

For RGB-only baselines, generated RGB frames are lifted using OTF. We also evaluate *Phys4D + WAN2.2 RGB+OTF*, where the RGB videos generated by Phys4D are lifted using the same off-the-shelf depth and flow estimators. The row *Phys4D + WAN2.2 native interface* uses learned Phys4D depth.

Metrics. We report depth estimation metrics: Absolute Relative Error (AbsRel), Root Mean Squared Error (RMSE), and threshold accuracies δ_1, δ_2, and δ_3. Lower AbsRel and RMSE indicate more accurate depth, while higher δ scores indicate better geometric correctness. Visual quality is evaluated separately in Sec. 6.3.

Table 21: **Per-frame 3D geometry.** The shared-extractor rows use the same frozen DAV2 depth estimator; the native-interface row uses Phys4D depth. No RGB tracker is used here.

Model	AbsRel ↓	RMSE ↓	δ_1 ↑	δ_2 ↑	δ_3 ↑
WAN2.2 + OTF	0.3929	0.9012	26.00	52.69	75.95
CogVideoX + OTF	0.3483	0.7854	32.45	68.97	84.35
Open-Sora V1.2 + OTF	0.4216	0.9485	23.84	49.73	72.18
Phys4D + WAN2.2 RGB+OTF	0.3347	0.7513	35.82	72.64	87.91
Phys4D + WAN2.2 native interface	**0.2386**	**0.5712**	**51.84**	**86.73**	**97.02**

Table 22: **Temporal geometry and motion consistency.** We report depth warp errors, RGB warp reconstruction errors, and optical-flow accuracy. **OTF** denotes off-the-shelf depth/flow lifting with DepthAnythingV2 (Yang et al., 2024a) and SEA-RAFT (Wang et al., 2024).

	Depth Warp		RGB Warp		Motion Dynamics		
Model	L1 ↓	Charb. ↓	LPIPS ↓	Charb. ↓	EPE ↓	Fl-all ↓	1px Out ↓
WAN2.2 + OTF	0.7990	0.7898	0.2409	0.0507	1.2516	6.2173	15.69
CogVideoX + OTF	0.7054	0.7893	0.2564	0.0527	1.2343	4.3243	12.34
Open-Sora V1.2 + OTF	0.7436	0.8534	0.2783	0.0589	1.2718	7.2343	18.96
Phys4D + WAN2.2 RGB+OTF	0.6698	0.6941	0.2216	0.0499	1.0824	3.8735	10.91
Phys4D + WAN2.2 native interface	**0.4558**	**0.4686**	**0.1637**	**0.0438**	**0.4478**	**1.1026**	**3.91**

Analysis. Table 21 shows that, with the external depth readout fixed, the matched WAN2.2 comparison reduces AbsRel from 0.3929 to 0.3347 and RMSE from 0.9012 to 0.7513. This is the RGB-derived structural comparison. The native-interface row reports a further AbsRel reduction from 0.3347 to 0.2386, which includes the specialized depth head and cannot be attributed solely to generated RGB. The pure-RGB quality columns are not used in this matched depth-readout comparison.

H.3 Temporal Geometry and Motion Consistency

Setting. This diagnostic evaluates whether geometry and motion remain locally consistent across adjacent frames. A model may produce individually plausible frames while still violating temporal coherence, for example by changing object depth, shape, or motion direction inconsistently across time. We therefore evaluate whether predicted depth, RGB appearance, and optical flow agree under temporal warping.

For the shared-extractor rows, depth and dense optical flow are obtained from generated RGB using the same OTF models. The native-interface row uses Phys4D's corresponding heads. All rows use the same held-out scene and camera protocol. The point tracker used for worldlines is not used for these per-pixel flow or adjacent-frame warping metrics.

Metrics. We report three groups of temporal metrics. **Depth Warp** measures the consistency of depth across adjacent frames by warping depth from time t to $t{+}1$ and comparing it against the target depth. We report L1 and Charbonnier errors. These evaluation values use an uncompensated transport error; unlike the GT-compensated training loss in Eq. (12), they can include genuine axial depth change. The depth-warp metric is therefore distinct from the training loss. **RGB Warp** measures appearance-level temporal consistency by warping RGB frames using predicted flow and comparing the result with the next frame using LPIPS and Charbonnier distance. **Motion Dynamics** evaluates optical flow using endpoint error (EPE), Fl-all, and 1px outlier rate.

Analysis. Table 22 shows that, with the OTF readout fixed, WAN2.2 Base and Phys4D RGB+OTF score 0.7990 and 0.6698 on Depth Warp L1, 1.2516 and 1.0824 on Flow EPE, and 0.2409 and 0.2216 on RGB Warp LPIPS. These matched-backbone differences describe structure and motion recoverable from generated RGB under shared estimators. In the separately reported native readout, the corresponding values are 0.4558, 0.4478, and 0.1637. This additional contrast includes the specialized depth and flow heads. These local warping and dense-flow metrics do not use the CoTracker3 worldline readout.

Table 23: **Full seven-method 4D diagnostics.** Worldline correspondences come from the same frozen CoTracker3 applied to each generated RGB video; tracked locations are lifted using the depth source of each row and compared with simulator trajectories. Chamfer and novel-time metrics use their separate protocols. OTF uses external DAV2/SEA-RAFT readouts; the Phys4D native row uses its depth head.

	Geometry	Trajectory / Worldline Dynamics					Novel Time	
Model	**Chamfer ↓**	**L2 Error ↓**	**Mean Drift ↓**	**Final Drift ↓**	**Fail Rate ↓**	**Length ↑**	**Depth Err ↓**	**Warp Err ↓**
WAN2.2 + OTF	0.5058	0.5367	0.5369	0.5576	12.38%	83.18	0.5841	1.1076
CogVideoX + OTF	0.4923	0.5239	0.5239	0.5348	11.34%	84.67	0.5432	1.1254
Open-Sora V1.2 + OTF	0.5286	0.5694	0.5637	0.5834	13.54%	80.24	0.6534	1.1573
Phys4D + WAN2.2 RGB+OTF	0.4873	0.5184	0.5126	0.5269	10.91%	85.42	0.5527	1.0954
PhysGen3D	0.4718	0.5026	0.4962	0.5148	10.24%	86.11	0.5325	1.0817
TesserAct	0.4596	0.4897	0.4819	0.5036	9.83%	86.94	0.5214	1.0679
Phys4D + WAN2.2 native interface	**0.4159**	**0.4375**	**0.4312**	**0.4556**	**8.36%**	**89.68**	**0.4746**	**1.0289**

H.4 Sequence-Level 4D World Evolution

Setting. This diagnostic covers three distinct tasks: set-level 4D geometric alignment, correspondence-aware worldlines, and novel-time continuity. Shared-extractor rows use OTF depth/flow; the native-interface row uses Phys4D heads. Simulator annotations supply the scoring references. Chamfer and novel-time metrics are computed separately and do not use the point-tracker correspondences.

4D geometry. We measure 4D Chamfer Distance between predicted and ground-truth spatio-temporal point sets. Each point is represented as (x, y, z, τ), where τ denotes normalized time. This fixed diagnostic is distinct from the locally time-relaxed Stage-III training reward in Appendix C.6; the two objectives have different matching rules.

Worldline dynamics. The RGB-derived protocol initializes a common query set $Q = \{\mathbf{q}_k\}_{k=1}^{N}$ in the shared conditioning frame and applies one frozen CoTracker3 point tracker (Karaev et al., 2024), denoted $\mathcal{K}$, to each method's complete generated video V^m:

$$\{(\hat{\mathbf{u}}^m_{t,k}, \hat{s}^m_{t,k})\}_{t,k} = \mathcal{K}(V^m, Q). \tag{35}$$

Here $\hat{s}$ denotes tracker visibility/validity. For readout $r \in \{\mathrm{ext}, \mathrm{native}\}$, valid tracked positions are lifted with the depth of that same readout and common camera intrinsics K:

$$\hat{\mathbf{X}}^{m,r}_{t,k} = \hat{D}^{m,r}_t(\hat{\mathbf{u}}^m_{t,k}) K^{-1} \widetilde{\hat{\mathbf{u}}}^m_{t,k}. \tag{36}$$

The two Phys4D readouts reuse the same V^m and identical 2D tracks; only their depth differs. All predicted and reference positions use the same fixed-camera coordinate frame. Simulator 3D trajectories associated with query ID k remain the references, compared at the same timestamps without Chamfer reassignment or test-time temporal alignment. GT flow, where used, constructs reference tracks only; it does not advance, repair, or reinitialize predicted tracks. Under the common visibility rule, GT-visible points lost in generated RGB enter failure statistics rather than disappear from the denominator. The five worldline metrics are computed from Eqs. (35)–(36) under one CoTracker3 evaluation protocol across methods. A fixed checkpoint, query set, depth-scale treatment, interpolation rule, failure thresholds, and aggregation are shared across the compared methods.

Novel-time continuity. We also test whether the lifted representation remains coherent at unseen timestamps. We split the sequence into observed and novel timestamps:

$$\mathcal{T}_{\mathrm{obs}} = \{0, 2, 4, \ldots\}, \qquad \mathcal{T}_{\mathrm{novel}} = \{1, 3, 5, \ldots\}. \tag{37}$$

Only observed frames are used to construct the partial world representation. For each novel timestamp, we interpolate geometry using flow-based warping:

$$\tilde{D}_{t+1} = \mathrm{warp}\Big(D^{\mathrm{pred}}_t,\, F^{\mathrm{pred}}_{t\to t+1}\Big). \tag{38}$$

The unseen target depth is used only to score the completed interpolation, not as an input to the transport step. We report Novel-Time Depth Error and Novel-Time Warp Error. Lower values indicate better temporal continuity.

Table 24: **Rigid-body consistency from RGB-tracked trajectories.** The protocol uses CoTracker3-derived 3D trajectories, GT mass, and simulator references on the rigid-body subset.

Model	Momentum Error ↓	Energy Error ↓
WAN2.2 + OTF	0.248	0.312
Phys4D + WAN2.2 RGB+OTF	0.223	0.284
Phys4D + WAN2.2 native interface	**0.176**	**0.221**

Analysis. Under the same external readout, WAN2.2 Base and Phys4D RGB+OTF score 0.5058 and 0.4873 on 4D Chamfer. The Phys4D native readout scores 0.4159; the larger readout gap includes specialized geometry prediction. On novel-time depth, the corresponding three scores are 0.5841, 0.5527, and 0.4746. These set-level and interpolation metrics are separate from point tracking. Under the common CoTracker3 readout, Mean Drift decreases from 0.5369 for WAN2.2 Base+OTF to 0.5126 for Phys4D RGB+OTF and 0.4312 for the Phys4D native-depth readout on the same Phys4D tracks. The corresponding failure rates are 12.38%, 10.91%, and 8.36%. These correspondence-aware scores remain subject to tracker errors and do not guarantee object identity.

H.5 Additional 4D Diagnostic Comparisons

The PhysGen3D and TesserAct rows are included in the shared seven-method Table 23 in Appendix H.4. They use the same held-out simulation diagnostics as the other rows there. Their input interfaces differ from the matched WAN2.2 base/Phys4D pair. All seven methods use the same frozen CoTracker3 on their own generated videos, and tracked locations are lifted with the corresponding depth readout.

Analysis. Compared with TesserAct, the Phys4D native-interface row reduces the set-level 4D Chamfer from 0.4596 to 0.4159 and CoTracker3 worldline Mean Drift from 0.4819 to 0.4312. These scores compare complete pipelines with different native interfaces; tracker error affects the Mean Drift comparison.

The separately defined novel-time errors are 0.5214 versus 0.4746 for depth and 1.0679 versus 1.0289 for warp, comparing TesserAct with Phys4D. These are interpolation diagnostics, not point-tracker scores.

H.6 Rigid-Body Physical Consistency

The 4D Chamfer and worldline metrics are general and can be applied across rigid, deformable, fluid, and coupled dynamics. However, some physical quantities, such as momentum and kinetic energy, are only well-defined when object mass and rigid-body trajectories are available. We therefore additionally evaluate known-mass momentum and energy reference errors on the rigid-body subset; these errors do not independently verify conservation laws in every scene.

Setting. For rigid-body scenes, velocities are reconstructed from the RGB-tracker-derived, depth-lifted trajectories of Appendix H.4, restricted to the rigid-body subset. The same tracker, query, depth-readout, and visibility rules apply. Simulator GT mass and trajectories remain the scoring references for momentum and kinetic-energy errors. This is a known-mass, simulator-reference diagnostic, not a universal conservation-law test or training reward.

Analysis. Table 24 reports momentum-error values of 0.248, 0.223, and 0.176 for WAN2.2 Base+OTF, Phys4D RGB+OTF, and Phys4D native interface; the corresponding energy errors are 0.312, 0.284, and 0.221. The native-interface row has the lowest reference error on this known-mass subset; the metric does not by itself establish conservation in unconstrained generated scenes.

I THEORY OF STAGE III RL

I.1 DENOISING AS A MARKOV DECISION PROCESS

Stage III optimizes a sample-level reward defined on the completed generated video. Following prior work on policy optimization for diffusion and flow-based generative models (Wallace et al., 2023; Liu et al., 2025), we cast the reverse denoising process as a finite-horizon Markov Decision Process.

Let K_{den} denote the number of reverse denoising transitions, indexed in descending order as $k = K_{\text{den}}, \dots, 1$. At reverse denoising step k, the state is

$$s_k = (c, k, \mathbf{z}_k), \tag{39}$$

where c denotes the conditioning input and $\mathbf{z}_k$ is the current video latent. The action is the next latent state,

$$a_k = \mathbf{z}_{k-1}, \tag{40}$$

and the policy is induced by the stochastic denoising model:

$$\pi_{\theta,\phi}(a_k \mid s_k) = p_{\theta,\phi}(\mathbf{z}_{k-1} \mid \mathbf{z}_k, c). \tag{41}$$

Here θ denotes the frozen pretrained video backbone and ϕ denotes the lightweight physics adapter parameters initialized from Stage II.

The final denoising transition occurs at $k = 1$, producing the clean latent $\mathbf{z}_0$ from $\mathbf{z}_1$. After decoding $\mathbf{z}_0$ into the completed video V, we assign the terminal reward to this transition:

$$r(s_k, a_k) = \begin{cases} R_{4D}(V), & k = 1, \\ 0, & k = 2, \dots, K_{\text{den}}. \end{cases} \tag{42}$$

The process terminates at $s_0 = (c, 0, \mathbf{z}_0)$, with no further action. This formulation allows Stage III to optimize sequence-level 4D alignment of the completed video rather than local per-frame losses.

I.2 STOCHASTICIZING FLOW-BASED SAMPLING

Flow-matching video models define deterministic sampling trajectories through a probability flow ODE:

$$d\mathbf{z}_\tau = v_{\theta,\phi}(\mathbf{z}_\tau, \tau, c)\, d\tau, \tag{43}$$

where $v_{\theta,\phi}$ is the velocity field induced by the frozen backbone θ and trainable adapters ϕ. Deterministic sampling provides no policy exploration, so Stage III uses a stochasticized flow for policy optimization.

Time convention. Appendix C.4 parameterizes the probability path by the interpolation time $s \in [0, 1]$ running from noise (s=0) to data (s=1), with target velocity $u_s = \mathbf{z}_1 - \mathbf{z}_0$ (Eq. (15)). This appendix instead adopts the standard *noising-time* convention $\tau = 1 - s$, in which τ=0 is clean data and τ=1 is pure noise, so that $\mathbf{z}_\tau = (1 - \tau)\, \mathbf{z}_{\text{data}} + \tau\, \boldsymbol{\epsilon}$ and the velocity appearing in Eqs. (43)–(46) is $v = d\mathbf{z}_\tau / d\tau = -u_s$, i.e., the same vector field with reversed sign, while reverse-time sampling proceeds from τ=1 to τ=0. Under this mapping the score–velocity identity, the sign of the SDE score correction, and the Euler–Maruyama transition below coincide exactly with the standard Flow-GRPO formulation (Liu et al., 2025); the two appendices describe the same network and the same sampler, and no reparameterization of the model is involved.

Following the probability-flow formulation, we introduce an SDE with the same marginal densities:

$$d\mathbf{z}_\tau = \left(v_{\theta,\phi}(\mathbf{z}_\tau, \tau, c) - \frac{1}{2} g^2(\tau) \nabla \log q_\tau(\mathbf{z}_\tau) \right) d\tau + g(\tau) d\mathbf{w}_\tau, \tag{44}$$

where q_τ is the marginal density, $g(\tau)$ controls the stochasticity, and $\mathbf{w}_\tau$ is a Wiener process.

For rectified-flow models, we use the score–velocity identity

$$\nabla \log q_\tau(\mathbf{z}_\tau) = -\frac{1}{\tau} \left(\mathbf{z}_\tau + (1 - \tau) v_{\theta,\phi}(\mathbf{z}_\tau, \tau, c) \right). \tag{45}$$

Substituting Eq. (45) into Eq. (44) and choosing $g(\tau) = \sigma_\tau$ yields

$$d\mathbf{z}_\tau = \left[v_{\theta,\phi}(\mathbf{z}_\tau, \tau, c) + \frac{\sigma_\tau^2}{2\tau} \left(\mathbf{z}_\tau + (1-\tau) v_{\theta,\phi}(\mathbf{z}_\tau, \tau, c) \right) \right] d\tau + \sigma_\tau d\mathbf{w}_\tau. \tag{46}$$

Discretizing Eq. (46) gives a Gaussian transition:

$$\pi_{\theta,\phi}(\mathbf{z}_{k-1} \mid \mathbf{z}_k, c) = \mathcal{N}\left(\mathbf{z}_{k-1}; \mu_{\theta,\phi}(\mathbf{z}_k, k, c), \sigma_k^2 \Delta\tau_k \, \mathbf{I} \right), \tag{47}$$

where $\mu_{\theta,\phi}$ is the Euler–Maruyama mean and $\Delta\tau_k > 0$ is the magnitude of the reverse-time step. Only non-degenerate stochastic transitions with positive variance enter PPO log-probability calculations.

I.3 Policy Optimization with Prior-Preserving KL

Stage III optimizes the 4D reward while preserving the real-video generative prior. Let $\pi_{\theta,\phi_{S2}}$ denote the reference Stage-II policy and $\pi_{\theta,\phi}$ the updated policy. The regularized Stage-III objective is

$$\max_{\phi} \; \mathbb{E}_{V \sim \pi_{\theta,\phi}} \left[R_{4D}(V) - \beta \mathbb{E}_k \, \overline{\mathrm{KL}} \left(\pi_{\theta,\phi}(\cdot \mid s_k) \,\|\, \pi_{\theta,\phi_{S2}}(\cdot \mid s_k) \right) \right]. \tag{48}$$

Here $\overline{\mathrm{KL}}$ is the KL per valid latent dimension and the expectation averages over optimized denoising steps. The Stage-II reference is a regularizer, not an additional physical reward.

We use a current policy $\pi_{\theta,\phi}$, a rollout policy $\pi_{\theta,\phi_{\mathrm{old}}}$ fixed during each collection round, and the reference policy $\pi_{\theta,\phi_{S2}}$ fixed throughout Stage III. For a sampled transition $a_{g,k}$ with m_k valid latent dimensions, let $\ell_{\phi,g,k} = m_k^{-1} \log \pi_{\theta,\phi}(a_{g,k} \mid s_{g,k})$. The latent-dimension-normalized ratio surrogate is

$$\widetilde{\rho}_{g,k}(\phi) = \exp(\ell_{\phi,g,k} - \ell_{\phi_{\mathrm{old}},g,k}). \tag{49}$$

It is not the full joint importance ratio. The corresponding reference KL is

$$\overline{\mathrm{KL}}_{g,k} = \frac{1}{m_k} D_{\mathrm{KL}}(\pi_{\theta,\phi}(\cdot \mid s_{g,k}) \,\|\, \pi_{\theta,\phi_{S2}}(\cdot \mid s_{g,k})). \tag{50}$$

Using the video-level advantage $\hat{A}_g$ defined below, the PPO-style surrogate and minimized training loss are

$$\mathcal{J}_{\mathrm{PPO}} = \widehat{\mathbb{E}}_{g,k} \left[\min \left(\widetilde{\rho}_{g,k} \hat{A}_g, \mathrm{clip}(\widetilde{\rho}_{g,k}, 1-\epsilon_{\mathrm{PPO}}, 1+\epsilon_{\mathrm{PPO}}) \hat{A}_g \right) \right], \tag{51}$$

$$\mathcal{L}_{StageIII} = -\mathcal{J}_{\mathrm{PPO}} + \beta \, \widehat{\mathbb{E}}_{g,k} \overline{\mathrm{KL}}_{g,k}. \tag{52}$$

Both terms average over the same sampled states and optimized denoising steps. The surrogate is computed on actions sampled by $\pi_{\theta,\phi_{\mathrm{old}}}$; the reference policy does not generate separate actions for this ratio.

Advantage estimation and per-step assignment. Stage III uses the importance-sampling PPO estimator of DDPO (Black et al., 2023), which by design has no value network. For each paired simulation scene s (the *context*: initial frame, prompt, and simulator rollout), we draw a group of G denoising trajectories and normalize the terminal 4D reward within the context by its mean and standard deviation,

$$\begin{aligned} \mu_s &= G^{-1} \sum_{g=1}^{G} R_{4D}(V_g; s), \\ \sigma_s^2 &= G^{-1} \sum_{g=1}^{G} (R_{4D}(V_g; s) - \mu_s)^2, \\ \hat{A}(V_g) &= \mathrm{clip}\left(\frac{R_{4D}(V_g; s) - \mu_s}{\sigma_s + \epsilon_A}, -5, 5 \right). \end{aligned} \tag{53}$$

and assign the same value to every optimized denoising step of that video. The subtracted per-context mean μ_s is the baseline; this is the "normalized advantage" referred to in Appendix D.3. The per-step log-probabilities entering $\widetilde{\rho}_{g,k}$ are computed analytically from the Gaussian transition in Eq. (47).

Trainable parameters and reward evaluator. During Stage III, the pretrained video backbone θ remains frozen. Only the lightweight physics adapters ϕ initialized from Stage II are updated. The locally time-relaxed geometric alignment is used as a scalar reward for policy optimization; it is not backpropagated as a differentiable supervised loss through the video backbone. The depth and motion heads and learned gate parameters are also frozen. This prevents direct updates to the reward evaluator, although generated inputs to it can still change.

J Extended Related Work

J.1 Large-Scale Video Generation

Early video diffusion models established scalable spatiotemporal denoising and latent-space generation for increasingly long and high-resolution videos (Ho et al., 2022; Blattmann et al., 2023). Sora scales patch-based diffusion-transformer training across heterogeneous visual data and explicitly discusses both long-range coherence and remaining physics failures, making it representative of the promise and current limits of frontier video generation (Liu et al., 2024b). Movie Gen extends this frontier to a cast of media foundation models for video, audio, editing, and personalization (Polyak et al., 2024). CogVideoX, HunyuanVideo, and Wan systematize open large-scale training pipelines for high-quality text-to-video synthesis, showing that open and closed models alike can now provide very strong visual priors (Yang et al., 2025; Tencent Hunyuan Foundation Model Team, 2024; Wan et al., 2025). Phys4D is designed to preserve these real-video generative priors while improving their physical structure.

J.2 Physics-Aware Generation

Recent work has started to address physics more directly in video generation. DiffPhy injects physical rules into video diffusion via LLM/MLLM-based reasoning and verification (Zhang et al., 2025a). VideoREPA distills physics-sensitive relational structure from video understanding foundation models into text-to-video generators (Zhang et al., 2025b). Motion Forcing explicitly decouples point, shape, and appearance and uses masked point recovery to encourage stronger motion dynamics reasoning (Xu et al., 2026). PhysRVG formulates physics-aware reinforcement learning for video generative models, emphasizing collision-grounded feedback and post-training alignment (Zhang et al., 2026). NewtonGen introduces neural Newtonian dynamics to improve controllability and physically consistent motion synthesis (Yuan et al., 2025). PhysCtrl learns physics-grounded 3D point trajectories conditioned on material parameters and external forces, then uses them to drive image-to-video generation (Wang et al., 2025). RealWonder and WonderPlay combine physics simulation with action-conditioned generation, using simulation as an intermediate representation for controllable dynamic outcomes (Liu et al., 2026; Li et al., 2025). AVID and GameFactory further show how pretrained video diffusion models can be adapted into action-conditioned world models, although they are not primarily aimed at dense physics supervision (Rigter et al., 2024; Yu et al., 2025). Compared with these approaches, Phys4D emphasizes prior-preserving transfer of simulation-derived *structure*—rather than post-hoc control, task-specific rules, or simulator-facing action interfaces.

J.3 Relation to Action-Conditioned Simulation-in-the-Loop Systems

PhysGen3D, WonderPlay, PerpetualWonder, and RealWonder combine explicit 3D/4D scene representations with physics simulation for action-conditioned interaction (Chen et al., 2025a; Li et al., 2025; Zhan et al., 2026; Liu et al., 2026), and PhysCtrl predicts force- and material-conditioned 3D point trajectories to control image-to-video generation (Wang et al., 2025). Both these systems and Phys4D use simulation, but they differ along four axes. *(1) Task and inference setting.* RealWonder receives temporally varying 3D forces, force fields, robot commands, or camera actions, reconstructs a simulatable 3D scene, and runs physics for each test instance; Phys4D improves ordinary image- and text-conditioned future generation and requires no explicit action, reconstructed scene, or external solver at inference. *(2) Role and scale of simulation.* There, simulation is an online intermediate step that computes the consequence of one action; Phys4D uses simulation as an offline physical teacher whose asynchronous coupled-physics rollouts expand 200 templates into about 250K environments and 1.25M videos, whose dense structural annotations are distilled into model parameters, after which

the simulator is removed entirely. *(3) Learning formulation.* Reconstruction–simulation–rendering systems solve physics per scene; Phys4D's representation-to-local-to-global post-training outputs a generator carrying a reusable physical prior rather than a renderer following one simulator rollout. *(4) Central objective.* Those systems emphasize explicit action control, real-time interaction, or interactive 3D/4D reconstruction and simulation; Phys4D asks how simulation-derived physical knowledge can be injected into a pretrained video generator without destroying its real-video appearance prior and open-domain generative diversity. The two lines are therefore complementary rather than interchangeable. A quantitative comparison with RealWonder under a shared protocol is given in Appendix F.6, and comparisons with PhysGen3D and TesserAct (Zhen et al., 2025) on the 4D diagnostics in Appendix H.5.

J.4 Physics-Oriented Evaluation

Video evaluation has quickly evolved from broad perceptual quality metrics to more targeted diagnostics. VBench introduced a comprehensive benchmark suite spanning visual quality, temporal quality, and motion-related factors (Huang et al., 2023). VBench++ broadened coverage to image-to-video and trustworthiness-oriented evaluation (Huang et al., 2024). VBench-2.0 further shifts toward intrinsic faithfulness and explicitly includes physics and commonsense as core dimensions (Zheng et al., 2025). EvalCrafter and FETV provide broader evaluation toolkits for large video models and fine-grained prompt categories (Liu et al., 2024a; 2023b). T2V-CompBench and DEVIL focus on compositional binding and dynamics-specific evaluation, respectively (Sun et al., 2025; Liao et al., 2024). VideoPhy and VideoPhy-2 evaluate whether generated videos obey physical commonsense, especially in material interactions and action-centric scenarios (Bansal et al., 2024; 2025). PhyGenBench explicitly organizes prompts around physical laws and domains such as mechanics, optics, thermal processes, and material properties (Meng et al., 2025). Physics-IQ goes further by asking whether video models truly understand physical principles rather than only appearing plausible (Motamed et al., 2025). Phys4D is complementary to these works: they provide valuable *diagnostics*, while Phys4D targets dense training supervision for physics-aware generation.

J.5 Geometry & Motion Estimation

Phys4D also relates to work on extracting geometry and correspondences from real videos. Depth Anything and Depth Anything V2 demonstrate strong open-world monocular depth estimation, while Video Depth Anything extends this line to temporally consistent depth over long videos (Yang et al., 2024b; Chen et al., 2025b). RAFT remains a standard reference point for dense optical flow estimation (Teed & Deng, 2020). TAPIR and CoTracker3 provide strong long-range point tracking and correspondence estimation, which are highly relevant for reasoning about persistent motion and visibility over time (Doersch et al., 2023; Karaev et al., 2024). These methods provide useful geometric or motion cues, but they do not by themselves inject physical dynamics into video generation. Phys4D instead uses geometry and motion as the interface through which physical structure is transferred into pretrained video diffusion models.

J.6 Dynamic 3D/4D Modeling

A separate line of work focuses on explicit representations for dynamic scenes. D-NeRF and HyperNeRF extended neural radiance fields to non-rigid and topologically varying scenes (Pumarola et al., 2021; Park et al., 2021). Dynamic 3D Gaussians and 4D Gaussian Splatting improved efficiency and rendering speed for dynamic view synthesis with explicit Gaussian-based representations (Luiten et al., 2023; Wu et al., 2024a). CAT4D converts monocular video into multi-view video using a diffusion model and reconstructs explicit 4D scenes from it (Wu et al., 2024b). Dynamic Point Maps propose a more feed-forward formulation for dynamic 4D geometry and motion prediction (Sucar et al., 2025). PhysGaussian integrates Newtonian dynamics into 3D Gaussians for generative motion synthesis (Xie et al., 2024). DreamPhysics uses video diffusion priors to optimize material properties for an MPM simulator (Huang et al., 2025a). PhysGen3D estimates amodal geometry, physical properties, and controllable initial conditions to create interactive 3D worlds from a single image (Chen et al., 2025a). WonderPlay closes the loop between a physics solver and a video generator for action-conditioned dynamic 3D scene generation (Li et al., 2025). These works largely target explicit 4D reconstruction or animation. Phys4D differs in using RGB-D-motion as a *training interface* for video diffusion, rather than requiring full explicit 4D scene generation at inference time.

J.7 Physical Interaction Datasets

Several datasets are important for dynamic scene understanding and physical interaction modeling. HOI4D provides large-scale egocentric RGB-D videos with rich annotations for human-object interaction (Liu et al., 2022). BEHAVE captures full-body human-object interaction with multi-view RGB-D data and 3D contact-aware fits in natural environments (Bhatnagar et al., 2022). InterCap extends markerless 3D human-object tracking with multi-view RGB-D capture and finer whole-body interaction detail (Huang et al., 2022). Ego-Exo4D provides large-scale multimodal ego/exo capture with video, point clouds, gaze, and pose annotations for skilled activities (Grauman et al., 2024). Panoptic Studio remains a foundational large-scale multi-view capture system for social motion capture (Joo et al., 2015). Neural 3D Video provides a standard multi-view dynamic scene benchmark for 3D video synthesis and dynamic view interpolation (Li et al., 2022). These datasets are essential for perception and reconstruction, but they do not by themselves provide scalable dense physical labels such as contacts, future trajectories, or simulator-side state variables across broad open-domain dynamics.

J.8 Simulation for Supervision

Simulation is especially attractive when dense physical structure is required. Kubric offers large-scale synthetic data generation with physics, rendering, and rich annotations in a unified pipeline (Greff et al., 2022). Isaac Sim provides a physically based virtual environment for simulation, testing, and synthetic data generation, while Isaac Gym and Isaac Lab emphasize large-scale GPU-accelerated physics for learning and policy training (NVIDIA Corporation, 2024; Makoviychuk et al., 2021; Mittal et al., 2025). GarmentLab provides a unified simulation and benchmark environment for garment and deformable-object manipulation, highlighting the value of multi-physics environments (Lu et al., 2024b). Domain randomization is a classic strategy for transferring from simulation to reality without overfitting to simulator appearance (Tobin et al., 2017). On the simulation side, PBD offers stable real-time constraint-based dynamics, MLS-MPM provides strong support for large deformation and contact-rich continuum simulation, and SoftMAC explicitly couples soft bodies, articulated rigid bodies, and clothes in a differentiable framework (Müller et al., 2007; Hu et al., 2018; Liu et al., 2023a). Phys4D follows a structure-focused, appearance-constrained adaptation principle in which simulation supervises *structure*—depth, motion, visibility, masks, and trajectories—while the pretrained generator remains anchored to real-video visual priors.

J.9 Diffusion Alignment and Adapters

Phys4D is also related to lightweight adaptation and alignment methods for diffusion models. LoRA introduced low-rank adaptation as an efficient way to inject new behavior into large pretrained models without full fine-tuning (Hu et al., 2021). ControlNet adds trainable spatial control branches while preserving the backbone diffusion prior (Zhang et al., 2023). T2I-Adapter similarly provides lightweight conditioning modules for structure-aware control (Mou et al., 2023). DDPO shows that diffusion models can be optimized directly for downstream reward objectives using reinforcement learning (Black et al., 2023). Phys4D is aligned with this general philosophy of preserving pretrained priors while adding targeted supervision, but its goal is not generic controllability or preference alignment; it is structure-focused, appearance-constrained transfer of physical knowledge into video diffusion models.

K Physics Simulation

Based on Isaac Sim (NVIDIA Corporation, 2024; Lu et al., 2024b), our simulation pipeline supports a broad range of object categories and physical phenomena to enable fine-grained physical interactions. We adopt *tailored simulation methods* according to object-specific physical characteristics: rigid and articulated objects are modeled with rigid-body dynamics, large garments and fluid materials are simulated using *Position-Based Dynamics* (Bender et al., 2015), and small elastic garments (e.g., gloves and socks) as well as deformable everyday objects (e.g., toys and sponges) are simulated using the *Finite Element Method* (Ciarlet, 2002). Additional object types such as ropes, inflatables, granular materials, and thermodynamic effects are handled by appropriate physics solvers, collectively exposing the model to diverse fine-grained physical regimes.

K.1 Simulation Methodology

Our simulation pipeline incorporates multiple physics solvers to model diverse physical phenomena encountered in complex interactive scenes, including cloth, fluids, deformable objects, rigid bodies, and environmental flow. Each component is selected to balance physical realism, numerical stability, and large-scale data generation efficiency.

K.1.1 Cloth Simulation via Position-Based Dynamics

We simulate garments using a Position-Based Dynamics (PBD) formulation, which represents cloth as a set of particles connected by geometric constraints. The cloth surface is discretized into a triangular mesh, where each vertex corresponds to a particle with associated position, velocity, and mass properties. Physical behavior emerges from enforcing constraints that regulate stretching, bending, and collisions.

Stretching constraints preserve local distances between neighboring particles, preventing excessive elongation, while bending constraints maintain angular relationships across adjacent mesh elements to model resistance to folding. Collision constraints ensure non-penetration between cloth particles and other objects in the scene. Rather than integrating forces explicitly, PBD iteratively projects particle positions to satisfy these constraints at each time step, yielding stable and efficient cloth dynamics even under large deformations.

K.1.2 Fluid Simulation with Particle-Based Constraints

Fluid behavior is simulated using a particle-based PBD framework, where each particle represents a small volume of fluid. The primary objective is to enforce approximate incompressibility while maintaining stable interactions with solid boundaries.

This is achieved by imposing density constraints that regulate local particle concentration, together with collision constraints that handle fluid–solid interactions. At each simulation step, particles are advanced under external forces, then iteratively corrected to satisfy density and collision constraints. This approach enables visually plausible fluid motion with high numerical stability, making it suitable for large-scale data generation.

K.1.3 Deformable Object Simulation via Finite Elements

For volumetric deformable objects, we employ a Finite Element Method (FEM) formulation that captures elastic and damping effects under external forces. Objects are discretized into tetrahedral meshes, with material behavior governed by elasticity, density, and damping parameters.

The simulation separates meshes used for visualization, collision detection, and physical computation, allowing efficient resolution of contact and deformation. By solving the equations of motion over finite elements, the simulator produces realistic volumetric deformations while maintaining controllable computational cost.

K.1.4 Environmental Flow and Wind Effects

To model environmental forces such as wind, we incorporate flow-based models that approximate the interaction between moving air and scene objects. These models capture directional flow, turbulence, and pressure-induced forces that influence both rigid and deformable bodies.

Wind effects are specified through configurable flow fields that define direction and magnitude, and their influence is applied consistently across interacting objects. This enables dynamic responses such as cloth fluttering and object deflection under varying environmental conditions.

K.1.5 Fire, Smoke, Magnetics and Optical

We simulate fire and smoke using fluid-based models to capture their evolving spatial structures and temporal dynamics. Magnetic interactions are modeled using physics-based magnetic-force equations, with the resulting forces incorporated into the dynamics of magnetically responsive objects. Optical effects are generated using RTX ray tracing to model light interactions with scene geometry and materials, producing visual observations that reflect the corresponding optical properties.

K.1.6 RIGID BODY DYNAMICS

Rigid objects are simulated using standard rigid body dynamics, where objects are assumed to maintain fixed internal geometry. Their motion is governed by linear and angular dynamics under external forces and collision impulses.

Collision handling relies on geometric approximations such as convex hulls or signed distance representations to balance accuracy and efficiency. Constraint solvers ensure stable resolution of contacts and joint limits, allowing reliable simulation of complex multi-object interactions.

K.2 PARAMETERIZATION AND PHYSICAL DIVERSITY

Rather than tuning individual parameters for specific outcomes, we adopt broad parameter ranges to encourage physical diversity across scenes. Within each physics module, parameters controlling stiffness, damping, friction, density, and interaction strength are randomized within physically plausible bounds.

This strategy enables the simulator to generate a wide spectrum of behaviors for cloth, fluids, deformable objects, and rigid bodies, while avoiding overfitting to narrow physical regimes. As a result, the collected data exhibits substantial variability in geometry, motion patterns, and interaction dynamics, which is critical for training robust models of observable geometry-motion dynamics.

Systematic Physical Parameterization. To disentangle physical phenomena for effective learning, we decompose generation into nine distinct categories (e.g., Rigid Body, Fluids, Soft Bodies, Thermodynamics). This supports **curriculum learning** where models progress from mastering individual laws to complex compositional scenarios. As detailed in Table 25, we enforce systematic randomization across material, environmental, and geometric attributes. We structure the datasets with progressive complexity (single to multi-object) (Bengio et al., 2009) and introduce parameter perturbation to prevent homogeneity, resulting in over 20,000 unique physical configurations.

Table 25: **Simulation Parameter & Curriculum Configuration.** We structure the dataset with progressive complexity and randomize physical attributes to ensure diverse and robust data generation.

Category	Parameters	Range / Configuration
Curriculum	**Complexity Dist.**	**Single (30%), Two-body (35%), Multi-object (35%)**
Material	Density (ρ), Young's Mod. (E) Poisson (ν), Friction ($\mu_{s,k}$) Restitution (e)	$\rho \in [100, 10\,000]$ kg/m^3, $E \in [10^6, 10^{11}]$ Pa $\nu \in [0.2, 0.5]$, $\mu_{s,k} \in [0.1, 1.0]$ $e \in [0.1, 0.95]$
Environment	Gravity (g), Lighting Variables	$g \in [9.2, 10]$ m/s^2, Intensity $\in [100, 100\,000]$ lux Air viscosity, Temperature gradients, 1000+ HDRIs
Geometry	Shape Types Scale (s), Aspect Ratio	Cubes, spheres, cylinders, complex meshes $s \in [0.1, 5.0]$ m, Randomized aspect ratios
Perturbation	Perturbation Ratio (δ)	$\delta \in [0.05, 0.25]$ (Randomized around base values)
Visuals	Reflectance (Roughness, Metallic) Appearance	Roughness $\in [0.05, 0.95]$, Metallic $\in [0, 1]$ Randomized RGB $\in [0, 1]^3$, Surface textures

L LIMITATIONS AND IMPACT STATEMENT

L.1 LIMITATIONS

We first state the limitations of the proposed method itself, and then the scope of the present study.

Dependence on simulator-paired rollouts. Stage II and Stage III draw all physical supervision from simulation, and Stage III aligns each generated sample to *one* simulator rollout under the paired initial state, camera, and physical parameters. Where the underlying dynamics are stochastic or under-determined by the initial frame, alignment to a single rollout may suppress other physically valid futures, and the learned prior may inherit solver-specific behavior (contact models, PBD/FEM approximations, and parameter ranges). Broad domain randomization and the frozen backbone mitigate but do not eliminate this bias.

Scope of the 4D alignment reward. The Stage-III reward measures set-level, time-indexed agreement of dynamic-region geometry within a fixed one-frame temporal neighborhood (Appendix C.6). Its local matches need not be globally monotone, and it does not establish persistent object identities. It is category-agnostic precisely because it does not presuppose persistent point identity, and it therefore does not by itself enforce identity-preserving worldlines, contact, momentum, or any explicit physical law; these are transferred only through their observable geometric consequences in the simulator rollouts. The worldline evaluation uses frozen CoTracker3 on generated videos to recover predicted correspondences, with simulator trajectories as references (Appendix H.4). Tracker errors and visibility limits affect these measurements, so improved worldline scores do not guarantee persistent object identity. The simulated diagnostics still rely on reference trajectories, calibration, and reference-side validity, and do not establish reference-free physical verification. The rigid-body momentum and energy diagnostic uses the same RGB-tracked trajectories together with GT mass (Table 24).

Scale of the RGB-D-motion interface. Depth is initialized with scale-normalized pseudo-labels in Stage I and calibrated to simulator metric depth in Stage II (Appendix C.3). On real images at inference, the metric scale of the exposed geometry is therefore inherited from the simulator calibration and may be biased for scenes whose scale statistics differ from the simulated ones; the exposed 4D quantities should be read as a consistent relative geometry-motion representation rather than as metrically exact reconstruction.

Residual use of simulator RGB. Simulator-rendered latents enter training through the high-noise flow-matching term. The controls in Sec. 6.4 and Appendix G.5 show that most of the reported Stage-II physics gain persists without simulator-video RGB flow matching, but they do not show that simulator appearance has zero influence at structure-forming denoising steps.

Generalization scope. The held-out evaluation tests generalization to unseen templates, initial states, and configurations within the supported physical regimes (Appendix A.1); it does not test unseen solvers or out-of-distribution physics. Gains are demonstrated on 1.1B–5B backbones; whether they persist, and whether returns diminish, on substantially stronger 16B–64B backbones remains untested (Appendix F.5).

Observable dynamics versus a reliable world model. Our evidence concerns improved *observable* video dynamics and geometry-motion consistency on external benchmarks and diagnostics. This should not be equated with a physically reliable world model suitable for robotics planning or scientific prediction, which would additionally require calibrated uncertainty, controllable interventions, and validated long-horizon behavior.

Scope of the study. The current study focuses on short-horizon video generation and evaluates physical consistency primarily through simulation-grounded supervision and benchmark protocols. Although the results show consistent gains across multiple backbones and external evaluations, extending the framework to substantially longer videos, broader real-world dynamics, and more open-ended interaction settings remains an important direction for future work. In particular, long-horizon generation may require stronger memory, compounding-error control, and more diverse supervision than the current short-video setting.

L.2 Impact Statement

This work aims to improve the physical consistency of generative video world models, which may benefit simulation, robotics, and scientific content creation by producing temporally and geometrically more coherent video predictions. At the same time, improved video generation quality can increase the risk of misuse in deceptive or misleading synthetic media. To mitigate this risk, we focus this submission on research disclosure and benchmark design rather than releasing a fully packaged public generation system, and we plan to accompany future asset release with documentation, licensing terms, and usage guidance.