ENACTPHYS: EXECUTING EVOLVING PHYSICAL PROCESSES IN VIDEO DIFFUSION

Anonymous authors
Paper under double-blind review

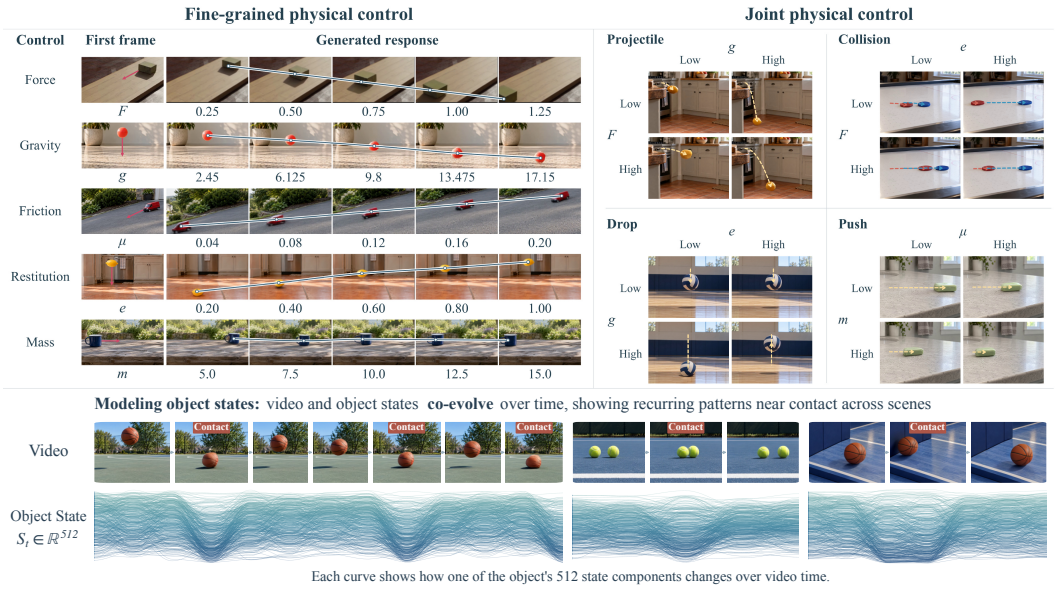

Figure 1: **EnactPhys generates distinct object motions as physical parameters vary individually or jointly. Upper:** Single-parameter sweeps (left) and paired-parameter grids (right) show separate generations from the same initial scene. **Lower:** EnactPhys achieves this controllability by modeling evolving object states S, whose recurring patterns around contact events across scenes reveal **interaction-relevant structure in the learned representation**.

ABSTRACT

Physical control in video generation requires translating user-specified parameters into corresponding motion responses. These responses arise through an evolving physical process in which parameter effects depend on the current state. Existing parameter-conditioning approaches leave this intermediate process largely implicit. Motivated by this task structure, we introduce **EnactPhys**, a video diffusion architecture for **state-mediated physical-process execution**. In our rigid-body instantiation, object states evolve through parameter-conditioned updates and inter-object interactions. We also introduce **PhysDelta**, a benchmark that jointly measures the **accuracy and selectivity** of physical control in simulation and matched real-world experiments. EnactPhys leads all four PhysDelta control metrics, improving over the strongest baseline for each metric by **15.0%–37.7%**, while maintaining competitive video quality and physical plausibility. It achieves the highest Verified Score on Physics-IQ (Solid Mechanics) among the evaluated methods, and ranks first in Invariance Score and second in Dynamical Score on MORPHEUS. State intervention experiments further show that learned state transformations transfer to held-out inputs and steer generated motion toward that induced by the target parameters, while keeping the original physical parameters fixed. Project page: `https://enactphys.github.io`.

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.

1 INTRODUCTION

Recent work explores video generation as a foundation for world models that support physical understanding, prediction, and interaction (Team Wan et al., 2025; Seedance et al., 2026; MiniMax, 2026; Ali et al., 2025; Agarwal et al., 2026). Such models should generate physically plausible motion consistent with scene constraints (Bansal et al., 2026; Meng et al., 2024; Motamed et al., 2026; Li et al., 2025a; Wang et al., 2026). Beyond producing plausible motion, controllable world models should allow users to specify the physical conditions governing that motion. Changing a physical parameter should then produce the expected change in the target object's motion and interactions.

Existing approaches guide video generation with motion scaffolds (Wang et al., 2025), force cues (Gillman et al., 2025), or object-level physical parameters (Narayanan et al., 2026; Li et al., 2026). Parameter-based methods encode physical conditions and use spatial alignment or object binding to specify where they apply.

Consider a ball pushed toward another by an external force. The force changes its motion, narrowing the gap between the balls and potentially leading to contact. If they collide, restitution influences their post-collision states and subsequent motion; otherwise, restitution does not affect their motion. This example illustrates how **the evolving scene state determines when and how physical parameters affect motion.** Existing parameter-conditioning approaches leave this process to be learned and executed implicitly by the video generator. Motivated by this task structure, we explicitly model and execute the physical process within video diffusion.

To this end, we introduce **EnactPhys**, a video diffusion architecture for **state-mediated physical-process execution**. Given an initial frame, object masks, and physical parameters, EnactPhys constructs object-state sequences over video time. In our rigid-body instantiation, object states evolve through temporal updates conditioned on force, gravity, and mass, while inter-object interactions are conditioned on friction and restitution. These states read from and write back to video features, influencing generated motion while supporting direct analysis and manipulation.

Existing benchmarks assess physical plausibility, adherence to physical laws, agreement with observed dynamics, and responses to interventions (Meng et al., 2024; Wang et al., 2026; Motamed et al., 2026; Begiristain et al., 2026; Cai et al., 2026). We introduce **PhysDelta** to systematically evaluate the **accuracy and selectivity** of physical control. It measures the accuracy of intended motion responses and the preservation of unaffected motion and non-target behavior. We directly measure motion under parameter interventions in simulation and matched real-world experiments.

EnactPhys leads all four PhysDelta control metrics by 15.0%–37.7% over the strongest baseline for each metric, with competitive video quality and physical plausibility. It achieves the highest Verified Score on Physics-IQ Verified (Solid Mechanics) (Motamed et al., 2026; Rädsch et al., 2026), and ranks first in Invariance Score and second in Dynamical Score on MORPHEUS (Tragoudaras et al., 2025). Architectural ablations and state interventions further examine how the proposed dynamics contribute to physical control.

Our main contributions are summarized as follows:

(1) We introduce EnactPhys, a video diffusion architecture that explicitly models and executes physical processes through object-state evolution and parameter-conditioned inter-object interactions.

(2) We introduce PhysDelta to jointly measure physical-control accuracy and selectivity under parameter interventions in simulation and matched real-world experiments.

(3) We provide intervention-based evidence for state-mediated physical control: learned state transformations generalize to held-out inputs and steer generated motion with the original physical parameters held fixed.

2 RELATED WORK

Controllable Video Generation. Controllable video generation aims to synthesize videos that follow user-specified conditions. Motion control guides camera or object movement through poses, trajectories, or point tracks (He et al., 2025; Wu et al., 2024; Geng et al., 2025). Physical control instead specifies forces and physical properties, allowing motion to emerge from these conditions. To realize such controls, simulation-guided methods derive motion from physics engines (Liu et al.,

2024; Xie et al., 2025; Li et al., 2025b). Learning-based alternatives predict motion or generate videos directly from physical conditions (Wang et al., 2025; Gillman et al., 2025; 2026). Among these, PhyCo and PhyParam condition video dynamics through spatial property maps and object-bound parameter tokens (Narayanan et al., 2026; Li et al., 2026), PHANTOM (Shen et al., 2026) jointly models video and latent physical dynamics. These approaches introduce physical parameters as conditions for video generation, leaving the intermediate physical process to be learned implicitly. EnactPhys explicitly models and executes this process within video diffusion.

Physical Benchmarks for Video Generation. Physical benchmarks assess whether generated motion is physically plausible and responds correctly to specified conditions. Physical plausibility benchmarks use human raters or learned evaluators to assess whether generated falls, collisions, and object interactions follow physical expectations (Meng et al., 2024; Bansal et al., 2026; Li et al., 2025a; Lin et al., 2026). Quantitative benchmarks use reference-based and intervention-based tests. Reference-based tests compare generated motion with reference dynamics or analytical laws (Motamed et al., 2026; Tragoudaras et al., 2025; Wang et al., 2026; Thozhiyoor et al., 2026; Li et al., 2026). Intervention-based tests compare generated motion under controlled changes in scene conditions and physical parameters (Begiristain et al., 2026; Khanbayov & Kurban, 2026; Cai et al., 2026). In particular, What-If World (Cai et al., 2026) evaluates intervention outcomes through paired VLM judgments, checking both the expected response and the preservation of unaffected behavior. PhysDelta goes further by measuring how faithfully generated motion follows the intended intervention: how accurately the targeted motion responds, and how well unaffected motion and non-target behavior are preserved. These complementary measurements jointly assess the accuracy and selectivity of physical control.

	Scale	Data	Evaluation					Coverage	Release
Benchmark	**Videos / model**	**Real-world scenes**	**Video quality**	**Phys. plaus.**	**Param. ctrl**	**Obj. ctrl**	**Unaffected motion**	**Physical variables**	**Eval. code**
WorldModelBench	350	✓	✓	✓	×	×	×	–	✓
Physics-IQ	198	✓	×	✓	×	×	×	–	✓
PhyGround	250	✓	✓	✓	×	×	×	–	✓
MORPHEUS	–	✓	✓	✓	×	×	×	–	✓
PhyParam-Bench	380	×	✓	✓	✓	×	×	F, m, μ, g, e	×
Principia	1,058	✓	✓	✓	✓	×	✓	h, m, I, l	×
PhysWeep	210	×	×	✓	✓	×	×	$g, \omega, \zeta, e, k, \mu$	✓
What-If World	638	✓	✓	✓	✓	✓	✓	Six intervention types	×
PhysDelta (ours)	**3,594**	✓	✓	✓	✓	✓	✓	$\boldsymbol{F, m, \mu, g, e}$	✓

Table 1: Comparison of video-generation benchmarks for physical evaluation.

3 METHOD

We introduce EnactPhys, a video diffusion architecture that explicitly models rigid-body dynamics through parameter-conditioned object-state evolution and inter-object interactions. We first describe the architecture, followed by the training data and objectives.

3.1 ARCHITECTURE

We build on a pretrained Wan (Team Wan et al., 2025) video diffusion Transformer (Peebles & Xie, 2023) (DiT) and add an object-state branch to explicitly model parameter-dependent physical processes (Fig. 2). The branch reads object-specific video features, updates object states through temporal evolution and inter-object interactions, and writes the updated state information back into video features to influence generated motion.

Object state modeling. We construct a compact latent object state space by aggregating and projecting object-specific DiT features. In this space, each object is represented by a sequence of learned states indexed by video time. Let $s_{i,t}$ denote the state of object i at time step t in the latent video sequence. Together, these states form $S = \{s_{i,t}\}$.

Object state evolution. We update object states through two complementary processes: temporal evolution within each object and interactions between objects.

Temporal evolution. For each object, we update its state sequence $S_i = (s_{i,1}, \ldots, s_{i,T})$ using separate embeddings of external force F_i, gravitational acceleration g_i, and mass m_i (Fig. 2(c)):

$$\widetilde{S}_i = \mathcal{T}\big(S_i + E_F(F_i) + E_g(g_i) + E_m(m_i)\big), \tag{1}$$

where E_F, E_g, E_m embed the physical conditions and $\mathcal{T}$ applies temporal attention within each object.

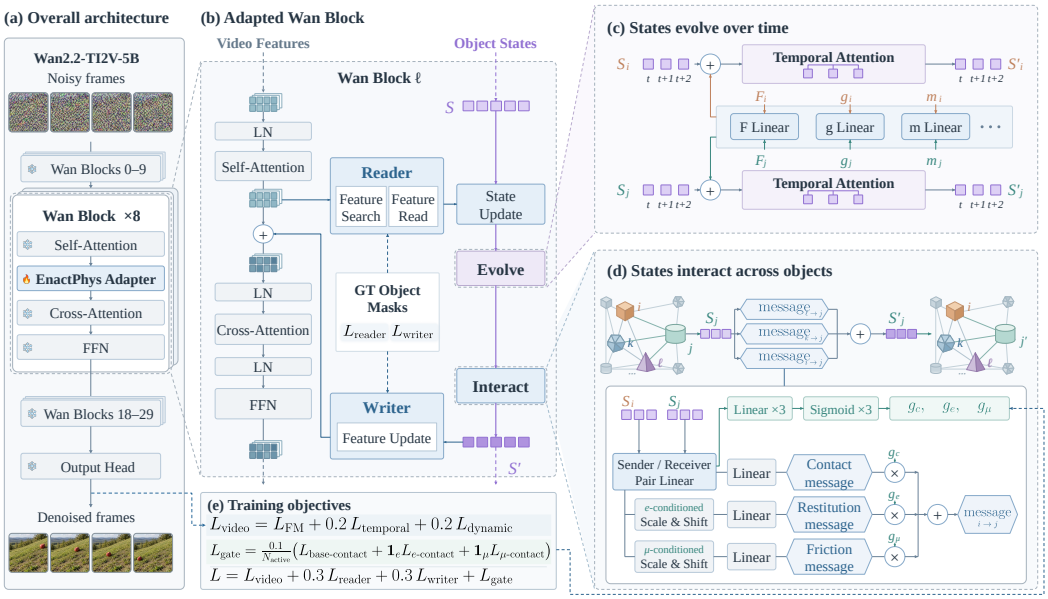

Figure 2: **Overview of the EnactPhys architecture.** Eight consecutive Wan blocks are augmented with an object-state branch. Conditioned on physical parameters, object states evolve over time and interact across objects in each adapted block. The Reader and Writer exchange information between object states and video features.

Inter-object interactions. At each time step, we compute messages in both directions between temporally updated object states (Fig. 2(d)). The message from object j to object i combines contact, restitution, and friction branches:

$$M_{ij} = g_c M^c_{ij} + g_e M^e_{ij} + g_\mu M^\mu_{ij}. \tag{2}$$

These branches model contact constraints, e-conditioned rebound, and μ-conditioned resistance to tangential motion, respectively. Independent learned gates g_c, g_e, g_μ weight their contributions, allowing messages to be suppressed in the absence of contact. Each object aggregates incoming messages from other objects and the support surface:

$$s'_i = \mathcal{U}\big(s_i, \textstyle\sum_{j \neq i} M_{ij}\big), \tag{3}$$

where $\mathcal{U}$ projects the aggregated message and refines the state through residual and feedforward updates.

Information exchange between video features and object states. The Reader updates existing object states with object-specific information from the current video features (Fig. 2(b)). After temporal evolution and inter-object interactions, the Writer writes the updated state information back to the corresponding object regions in the video features. Within each denoising pass, the updated object states are retained for subsequent updates.

3.2 Training Data and Objectives

Training data. We generate training videos using MuJoCo (Todorov et al., 2012) for rigid-body simulation and Blender Eevee (Blender Foundation) for rendering. We also include a subset of PhyCo (Narayanan et al., 2026) data and samples generated with its simulation pipeline. We vary physical conditions individually and jointly. The dataset contains 109,800 training samples and 4,196 validation samples, covering collisions, falling, sliding, motion on inclined planes, projectile motion, and balance scales. We collect per-frame object masks and labels for object–object and object–support contacts to supervise spatial localization and interaction gating.

Training objectives. We freeze the pretrained video backbone and train the added modules with

$$L = L_{\text{video}} + 0.3 L_{\text{reader}} + 0.3 L_{\text{writer}} + L_{\text{gate}}, \tag{4}$$

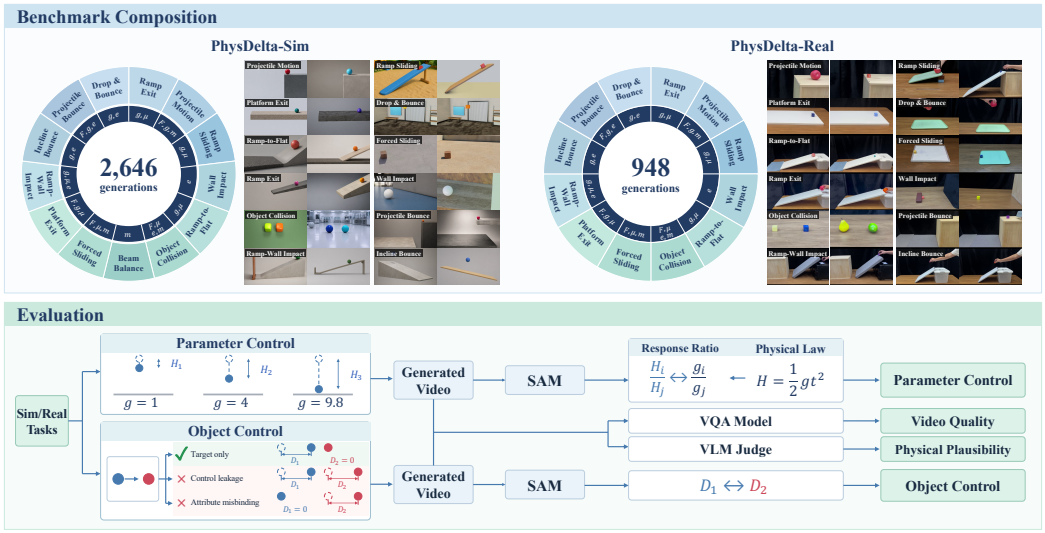

Figure 3: **Overview of PhysDelta. Upper:** PhysDelta-Sim and PhysDelta-Real cover matched physical-event families, with initial frames and associated parameters shown for each. **Lower:** PhysDelta systematically evaluates whether parameter changes produce the expected motion while preserving unaffected behavior through **Parameter Control** and **Object Control**; **Video Quality** and **Physical Plausibility** are assessed separately. Object tracking uses SAM 2 (Ravi et al., 2025).

where $L_{\text{video}} = L_{\text{FM}} + 0.2L_{\text{temporal}} + 0.2L_{\text{dynamic}}$ combines flow matching (Lipman et al., 2022), temporal-difference supervision (Gao et al., 2026), and motion-weighted prediction errors (He et al., 2026). Future-frame object masks penalize out-of-mask attention for Reader localization and Writer placement. Valid contact labels train three gates via weighted binary cross-entropy L_{gate}.

4 PhysDelta Benchmark

4.1 Motivation

Visual quality and plausible motion alone do not establish that a model responds correctly to specified physical parameters. **We posit that accurate physical control requires the expected changes in motion while preserving motion during unaffected phases and the behavior of unaffected objects.** For example, increasing an object's coefficient of restitution should produce a higher rebound without changing its pre-impact motion or the motion of other objects before their interaction.

Observed displacement depends on scene scale and viewpoint as well as physical parameters, so disagreement with a reference trajectory need not imply incorrect control. Building on this rationale and the principle above, we introduce PhysDelta, which compares generated motion across parameter settings within the same scene to assess expected changes and the preservation of invariance.

4.2 Benchmark Design

PhysDelta focuses on rigid-body dynamics involving five physical quantities: force F, gravitational acceleration g, coefficient of friction μ, coefficient of restitution e, and mass m. We design scenes around how these quantities affect motion, aiming for balanced coverage across the five physical dimensions. PhysDelta-Sim uses simulated or scene-constrained synthetic initial images. PhysDelta-Real uses initial frames from captured physical events to test whether physical control generalizes beyond synthetic scenes to real-world objects, backgrounds, and lighting.

Each task provides an initial image, a scene description, and physical parameter settings. Within each test group, we use the same initial image and vary one physical parameter while keeping the remaining physical parameters fixed, producing comparable generated videos. In tasks involving multiple objects, the parameter modification is applied only to the designated object.

4.3 Evaluation Metrics

Parameter Control. This metric evaluates whether physical parameter changes produce the expected motion. We measure motion quantities such as displacement, rebound height, or acceleration from generated videos. Let z_i, z_j denote the measurements under two parameter settings, and h_i, h_j the corresponding values predicted by the physics law. For direction and magnitude scoring, we apply a common normalization that maps the no-change ratio of 1 to 0:

$$\mathcal{N}(r) = \frac{r-1}{r+1}, \qquad r_{\text{measured}} = \mathcal{N}\left(\frac{z_j}{z_i}\right), \qquad r_{\text{expected}} = \mathcal{N}\left(\frac{h_j}{h_i}\right).$$

The direction score is

$$S_{\text{direction}} = \mathbf{1}[r_{\text{measured}} r_{\text{expected}} > 0].$$

The magnitude score is

$$S_{\text{magnitude}} = \begin{cases} 1 - \dfrac{(r_{\text{measured}} - r_{\text{expected}})^2}{r_{\text{measured}}^2 + r_{\text{expected}}^2}, & r_{\text{measured}} r_{\text{expected}} > 0, \\ 0, & \text{otherwise.} \end{cases}$$

The score subtracts the squared difference between the two ratios, normalized by their magnitudes, from 1. When the direction is correct, closer agreement in magnitude yields a higher score.

Separate test groups evaluate invariance in motion quantities that should remain unaffected by the parameter change:

$$S_{\text{invariance}} = \frac{\min(z_i, z_j)}{\max(z_i, z_j)}.$$

PhysDelta-Sim aggregates the three components. Our real-world data lack the numerical physical ground truth available in simulation, so PhysDelta-Real reports directional accuracy.

Object Control. This metric evaluates whether the designated object changes its motion as intended without unintended changes to other objects. We compare two generated videos that differ only in the target object's parameter setting. Let d_{target} and d_{other} denote the target and non-target motion differences between the paired videos. The factor $c \in [0, 1]$ measures how well the target object's motion change satisfies the expected physical response, with task-specific definitions given in Appendix C.3.

For valid target responses in the expected direction, the principal scoring form is

$$S_{\text{object}} = c\,\frac{d_{\text{target}}}{d_{\text{target}} + d_{\text{other}}}.$$

The score rewards the expected target response through c, which scales with response attainment up to one, while penalizing non-target changes. For rebound tasks in PhysDelta-Real, the target response is determined by the direction and size of the rebound-height change, while the score also penalizes changes to non-target trajectories.

Video Quality and Physical Plausibility. We evaluate video quality using FasterVQA (Wu et al., 2023) and physical plausibility using Qwen3-VL-235B-A22B-Instruct (Bai et al., 2025). The evaluator checks for visible anomalies such as object duplication, rigid-body deformation, and interpenetration. We report the proportion of videos that pass these checks. On a human calibration set of 204 videos with valid evaluation outputs, the revised evaluator achieved 91.67% agreement with human judgments and Cohen's $\kappa = 0.8082$.

5 Experiments

We evaluate EnactPhys through four research questions:

RQ1. Does EnactPhys improve control while preserving video quality and physical plausibility?
RQ2. Does physical control generalize to real-world scenes and novel event compositions?
RQ3. Beyond output-level results, how is physical control represented and realized?
RQ4. How do temporal evolution and inter-object interactions contribute to physical control?

5.1 Experimental Setup

Baselines. We evaluate general-purpose generators Wan2.2-TI2V-5B (Team Wan et al., 2025), CogVideoX-5B-I2V (Yang et al., 2025), and MiniMax H3 (MiniMax, 2026). We train three adaptations of Wan2.2-TI2V-5B: Wan (Text-only) uses object-bound structured parameter text; Wan (ControlNet) (Zhang et al., 2023) uses spatial parameter maps at object locations; and reproduced Phantom (Shen et al., 2026) extends physics-aware latent modeling with object-bound structured text for explicit physical control. We also evaluate prior explicit physical-control methods Force Prompting (Gillman et al., 2025) and PhyCo (Narayanan et al., 2026). Each baseline receives supported conditions through its native interface and unsupported conditions through the same structured-text format.

Implementation details. We train EnactPhys and the Wan2.2-TI2V-5B-based baselines for 6,000 steps on the same 109,800 videos (49 frames, 768×448). EnactPhys, Wan (Text-only), and Wan (ControlNet) use AdamW (Loshchilov & Hutter, 2019) with learning rate 10^{-4} and weight decay 0.01. Training uses 16 NVIDIA A100 80GB GPUs with batch size 2 per GPU (32 globally) and no gradient accumulation.

Evaluation. We evaluate physical control in rigid-body dynamics using PhysDelta, with the metrics defined in Section 4. We also evaluate on Physics-IQ Verified (Solid Mechanics) (Motamed et al., 2026; Rädsch et al., 2026) and MORPHEUS (Tragoudaras et al., 2025). Evaluation protocols and inference configurations are provided in Appendices C, D, and E.

5.2 Main Results

5.2.1 Physical Control and Generation Quality

Table 2 evaluates control accuracy and selectivity through Parameter Control and Object Control on PhysDelta-Sim and PhysDelta-Real, alongside Video Quality, Physical Plausibility, and external benchmark results.

Table 2: Comparison on PhysDelta, Physics-IQ Verified (Solid Mechanics), and MORPHEUS.

Control Paradigm	Models	PhysDelta-Sim / Real				Physics-IQ	MORPHEUS	
		Video Quality↑	Physical Plausibility↑	Parameter Control↑	Object Control↑	Verified Score↑	Dynamical Score↑	Invariance Score↑
General-purpose	Wan2.2-TI2V-5B	77.16 / 77.02	39.53 / 41.00	11.15 / 15.25	2.76 / 1.40	32.30	0.4515	0.4569
	CogVideoX-5B-I2V	70.10 / 72.31	66.98 / 48.00	15.96 / 38.98	12.38 / 22.76	34.10	0.5170	0.5072
	MiniMax H3	76.83 / 75.75	89.77 / 68.00	16.83 / 31.64	13.92 / 11.40	42.99	**0.8286**	0.7035
Latent physics control	Phantom	**79.84** / 78.75	92.56 / 86.00	48.00 / 74.44	24.54 / 28.20	46.22	0.6415	0.6213
Explicit physics control	Wan (Text-only)	78.72 / 78.20	**96.74** / 95.00	25.76 / 46.75	27.39 / 26.57	35.76	0.3915	0.4334
	Wan (ControlNet)	79.21 / 78.50	92.56 / 96.00	47.05 / 60.73	56.90 / 65.83	45.28	0.5511	0.5955
	Force Prompting	70.13 / 70.30	87.91 / 63.00	18.24 / 45.90	9.48 / 16.05	34.84	0.5410	0.5136
	PhyCo	67.88 / 70.56	94.42 / **98.00**	13.84 / 25.71	15.59 / 13.39	44.87	0.4425	0.6422
Ours	**EnactPhys**	79.56 / **79.18**	**96.74** / 97.00	**56.91 / 85.59**	**72.64 / 90.65**	**48.16**	0.6781	**0.7111**

EnactPhys leads all four PhysDelta control metrics (Table 2). Over the strongest baseline for each metric, Parameter Control and Object Control improve by 18.6% and 27.7% on PhysDelta-Sim, and 15.0% and 37.7% on PhysDelta-Real, with competitive video quality and physical plausibility. EnactPhys achieves the highest Verified Score on Physics-IQ Verified (Solid Mechanics) among the evaluated methods, and ranks first in Invariance Score and second in Dynamical Score on MORPHEUS. Figure 8 in Appendix A shows more distinct motion responses to parameter changes.

5.2.2 Generalization to Real Scenes and Composed Events

Real-world scenes. EnactPhys generalizes physical control to new objects and photographic scenes, including human subjects unseen during physical-control training, while maintaining coherent motion as parameters vary (Figure 4, left).

Event composition. It also shows compositional generalization (Figure 4, right). Training contains only separate projectile motion without ground contact and drop-and-bounce events, yet inference

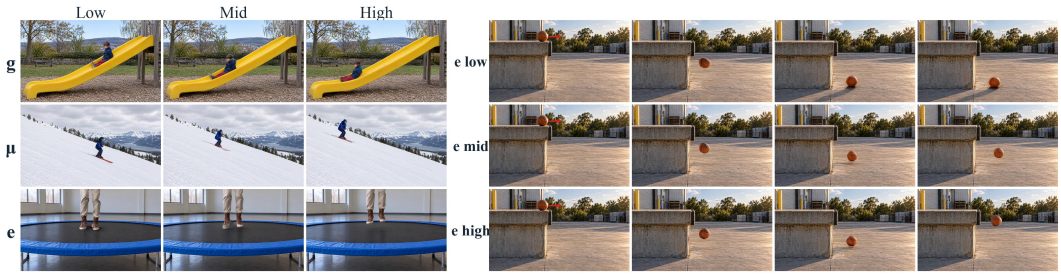

Figure 4: Generalization to photographic scenes and composed events. Left: varying g, μ, or e changes the generated motion, shown at matching times. Right: EnactPhys composes departure, fall, and rebound with e-controlled bounce height, while force and gravity remain fixed.

combines platform departure, fall, and rebound into a continuous sequence with controllable bounce height. See Figures 1 and 9 for additional real-world results.

5.2.3 What Object States Encode

Object states S encode object speed. We track object masks with SAM 2 and compute speed from their centroid trajectories. A linear probe fitted on S predicts this speed on held-out inputs without refitting, achieving $R^2 = 0.811$ and a mean absolute error of 1.48 pixels per frame across 360 videos from 12 held-out inputs. Predicted curves closely follow tracked speed over time (Figure 5), indicating that the temporal evolution of S reflects speed changes accessible through a linear readout. Each held-out input covers three physical settings and ten random seeds, with the same fitted probe used throughout.

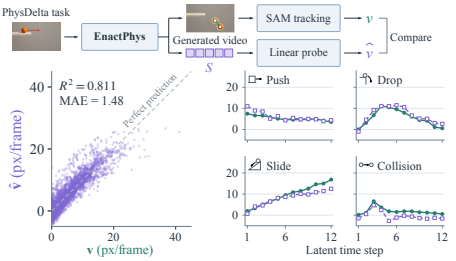

Figure 5: Object speed can be linearly decoded from object states.

Object states encode transferable changes for motion control. We fit an affine map between object states under two parameter values and apply it to held-out inputs. The transformed states closely match those obtained under the new parameter value, reducing mean state-prediction squared error by 79.9% relative to unchanged states. Normal generation retains the original parameters; State intervention applies the learned map with those parameters fixed; Parameter intervention changes the parameter itself (Figure 6). Using Parameter intervention as the reference, State intervention reduces mean trajectory squared error by 94.6% relative to Normal generation on held-out inputs.

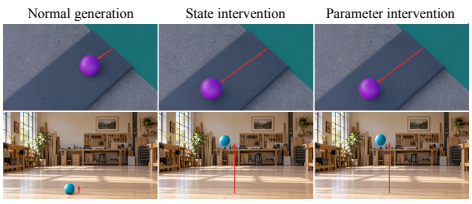

Figure 6: State intervention produces motion similar to parameter intervention.

5.2.4 When and How Physical Parameters Act

Physical parameters act within contact windows. Normal generation uses the base parameter value throughout; Full intervention uses the new value throughout; Windowed intervention uses the new value only during the observed contact interval and the base value elsewhere. Windowed and Full intervention produce closely matching motion (Figure 7A), indicating that the parameter-dependent response is concentrated within the contact window.

Physical parameters act through interaction messages. We vary one physical parameter while fixing all other inputs, measuring motion changes relative to the lowest parameter setting. We compare the original model with a version that retains the physical parameter inputs but zeros the corresponding restitution or friction message output. Removing these messages substantially flattens the

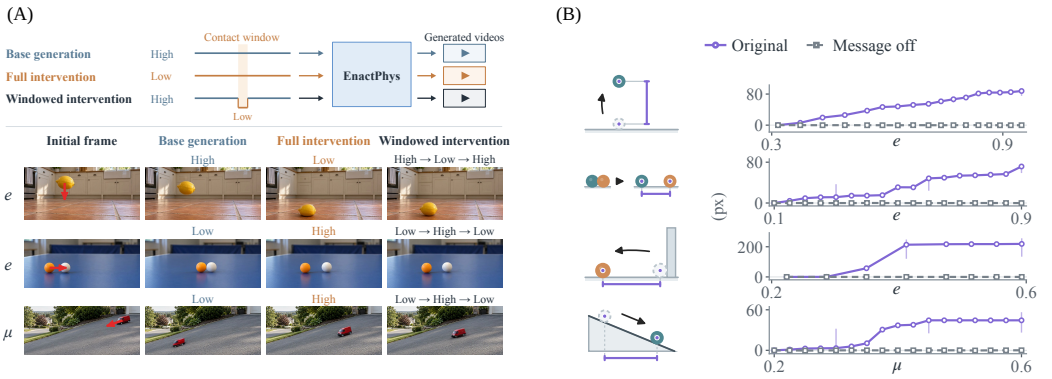

Figure 7: Physical control during interactions. (A) Contact-window interventions recover full-intervention effects. (B) Removing interaction messages weakens motion responses.

response curves (Figure 7B), identifying them as the pathway through which physical parameters affect generated motion.

5.3 Ablation Study

We compare four EnactPhys variants with the same Reader and Writer (Table 3):

(1) State Modeling Only. A linear mapping injects each object's physical conditions into its state, without temporal or inter-object updates.
(2) All-to-All State Attention. Object states attend to all objects at all time steps.
(3) w/o Temporal Evolution. We remove temporal attention but retain inter-object interactions.
(4) w/o Inter-object Interaction. We retain temporal attention and the motion-loss setting, replacing inter-object interactions with separate linear injections of μ and e.

State Modeling Only improves both control scores over Wan (Text-only), showing that object-state modeling benefits control even without temporal or inter-object updates. Parameter Control increases from 25.76 to 45.06 on Sim and from 46.75 to 73.31 on Real. Structured updates further improve Object Control over All-to-All State Attention from 42.03 to 72.64 (Sim) and from 49.09 to 90.65 (Real). Removing either temporal evolution or inter-object interactions reduces both control scores on both tracks, supporting their complementary roles.

Table 3: Ablation of object-state modeling and interaction structure on PhysDelta.

Variant	Object-State Modeling	State Interaction	PhysDelta-Sim		PhysDelta-Real	
			Parameter Control↑	Object Control↑	Parameter Control↑	Object Control↑
Wan (Text-only)	×	—	25.76	27.39	46.75	26.57
Wan (ControlNet)	×	—	47.05	56.90	60.73	65.83
State Modeling Only	✓	None	45.06	36.43	73.31	41.90
All-to-All State Attention	✓	All-to-all	47.71	42.03	79.52	49.09
EnactPhys	✓	Structured	**56.91**	**72.64**	**85.59**	**90.65**
w/o Temporal Evolution	✓	Inter-object only	54.34	48.29	77.40	56.02
w/o Inter-object Interaction	✓	Temporal only	49.40	53.84	77.68	45.78

6 Conclusions

EnactPhys introduces state-mediated physical-process execution into video diffusion through parameter-conditioned object-state evolution and inter-object interactions. It leads all four PhysDelta control metrics with competitive video quality and physical plausibility. Learned state transformations transfer to held-out inputs and steer generated motion at fixed physical parameters.

AI USE STATEMENT

Generative AI tools assisted with language polishing. A vision–language model was also used for Physical Plausibility scoring, as described in Section B. The authors reviewed the AI-assisted outputs and take responsibility for the final manuscript, code, and reported results.

ETHICS STATEMENT

Physically controllable video generation can support content creation and research, but realistic synthetic videos may also be used to misrepresent real events. We encourage clear identification of generated content and responsible use that respects privacy and the licenses of source assets and pretrained models.

REPRODUCIBILITY STATEMENT

The method and experimental setup are described in Sections 3 and 5. The appendix provides the evaluation protocol, baseline configurations, implementation details, training-data description, and state-intervention analyses (Sections C–G). We will release the code, model checkpoints, training data, and PhysDelta benchmark, together with evaluation configurations and scripts to reproduce the full quantitative comparison table.

REFERENCES

Niket Agarwal, Arslan Ali, Jon Allen, Martin Antolini, Adeline Aubame, Alisson Azzolini, Junjie Bai, Maciej Bala, Yogesh Balaji, Josh Bapst, et al. Cosmos 3: Omnimodal world models for physical ai. *arXiv preprint arXiv:2606.02800*, 2026.

Arslan Ali, Junjie Bai, Maciej Bala, Yogesh Balaji, Aaron Blakeman, Tiffany Cai, Jiaxin Cao, Tianshi Cao, Elizabeth Cha, Yu-Wei Chao, et al. World simulation with video foundation models for physical ai. *arXiv preprint arXiv:2511.00062*, 2025.

Shuai Bai, Yuxuan Cai, Ruizhe Chen, Keqin Chen, Xionghui Chen, Zesen Cheng, Lianghao Deng, Wei Ding, Chang Gao, Chunjiang Ge, et al. Qwen3-vl technical report. *arXiv preprint arXiv:2511.21631*, 2025.

Hritik Bansal, Clark Peng, Yonatan Bitton, Roman Goldenberg, Aditya Grover, and Kai-Wei Chang. Videophy-2: A challenging action-centric physical commonsense evaluation in video generation. In *International Conference on Learning Representations*, volume 2026, pp. 118456–118470, 2026.

León Begiristain, Olaf Dünkel, and Adam Kortylewski. Cronos: Benchmarking counterfactual physical consistency in video models. *arXiv preprint arXiv:2605.23699*, 2026.

Blender Foundation. Eevee. Blender Developer Documentation. URL `https://developer.blender.org/docs/features/eevee/`.

Kunlin Cai, Rui Song, Jinghuai Zhang, Kaiyuan Zhang, Pranav Bodapati, Alicia Yu, Fnu Suya, Mohammad Rostami, Jiaqi Ma, and Yuan Tian. What-if world: A causal benchmark for general world models in embodied scenarios. *arXiv preprint arXiv:2605.27589*, 2026.

Shenyuan Gao, William Liang, Kaiyuan Zheng, Ayaan Malik, Seonghyeon Ye, Sihyun Yu, Wei-Cheng Tseng, Yuzhu Dong, Kaichun Mo, Chen-Hsuan Lin, et al. Dreamdojo: A generalist robot world model from large-scale human videos. *arXiv preprint arXiv:2602.06949*, 2026.

Daniel Geng, Charles Herrmann, Junhwa Hur, Forrester Cole, Serena Zhang, Tobias Pfaff, Tatiana Lopez-Guevara, Yusuf Aytar, Michael Rubinstein, Chen Sun, et al. Motion prompting: Controlling video generation with motion trajectories. In *2025 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 1–12. IEEE, 2025.

Nate Gillman, Charles Herrmann, Michael Freeman, Daksh Aggarwal, Evan Luo, Deqing Sun, and Chen Sun. Force prompting: Video generation models can learn and generalize physics-based control signals. In D. Belgrave, C. Zhang, H. Lin, R. Pascanu, P. Koniusz, M. Ghassemi, and N. Chen (eds.), *Advances in Neural Information Processing Systems*, volume 38, Main Conference, pp. 103174–103201. Curran Associates, Inc., 2025. doi: 10.52202/085713-3448. URL https://proceedings.neurips.cc/paper_files/paper/2025/file/953dcf20cbd275465066ad63ea111f13-Paper-Conference.pdf.

Nate Gillman, Yinghua Zhou, Zitian Tang, Evan Luo, Arjan Chakravarthy, Daksh Aggarwal, Michael Freeman, and Chen Sun. Goal force: Teaching video models to accomplish physics-conditioned goals. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 20077–20087, 2026.

Hao He, Yinghao Xu, Yuwei Guo, Gordon Wetzstein, Bo Dai, Hongsheng Li, and Ceyuan Yang. Cameractrl: Enabling camera control for video diffusion models. In *International Conference on Learning Representations*, volume 2025, pp. 100433–100464, 2025.

Haoran He, Yang Zhang, Liang Lin, Zhongwen Xu, and Ling Pan. Pre-trained video generative models as world simulators. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 40, pp. 4645–4653, 2026.

Rasul Khanbayov and Hasan Kurban. Physweep: Does a video generator realize the physics you ask for? *arXiv preprint arXiv:2609.06207*, 2026.

Dacheng Li, Yunhao Fang, Yukang Chen, Shuo Yang, Shiyi Cao, Justin Wong, Michael Luo, Xiaolong Wang, Hongxu Yin, Joseph Gonzalez, Ion Stoica, Song Han, and Yao Lu. Worldmodelbench: Judging video generation models as world models. In D. Belgrave, C. Zhang, H. Lin, R. Pascanu, P. Koniusz, M. Ghassemi, and N. Chen (eds.), *Advances in Neural Information Processing Systems*, volume 38, Main Conference. Curran Associates, Inc., 2025a. doi: 10.52202/085713-1834. URL https://proceedings.neurips.cc/paper_files/paper/2025/file/4ec03ed08a3fcb59e1c815b5598beff1-Paper-Datasets_and_Benchmarks_Track.pdf.

Yanxun Li, Hao Wen, Bingze Song, Jiashu Zhu, Aiming Hao, Chubin Chen, Jintao Chen, Jiahong Wu, Xiangxiang Chu, and Miao Wang. Learning explicit physical parameter control and benchmarking for video generation. *arXiv preprint arXiv:2607.18924*, 2026.

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 *2025 IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 9080–9090. IEEE, 2025b.

Juyi Lin, Arash Akbari, Yumei He, Lin Zhao, Haichao Zhang, Arman Akbari, Xingchen Xu, Zoe Y Lu, Enfu Nan, Hokin Deng, et al. Phyground: Benchmarking physical reasoning in generative world models. *arXiv preprint arXiv:2605.10806*, 2026.

Yaron Lipman, Ricky TQ Chen, Heli Ben-Hamu, Maximilian Nickel, and Matt Le. Flow matching for generative modeling. *arXiv preprint arXiv:2210.02747*, 2022.

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

Ilya Loshchilov and Frank Hutter. Decoupled weight decay regularization, 2019. URL https://arxiv.org/abs/1711.05101.

Fanqing Meng, Jiaqi Liao, Xinyu Tan, Wenqi Shao, Quanfeng Lu, Kaipeng Zhang, Yu Cheng, Dianqi Li, Yu Qiao, and Ping Luo. Towards world simulator: Crafting physical commonsense-based benchmark for video generation. *arXiv preprint arXiv:2410.05363*, 2024.

MiniMax. MiniMax H3: An open model breaking the boundaries between tasks and modalities. MiniMax Research, July 2026. URL https://www.minimax.io/blog/minimax-h3. Official research blog, accessed August 20, 2026; technical report forthcoming.

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

Sriram Narayanan, Ziyu Jiang, Srinivasa Narasimhan, and Manmohan Chandraker. Phyco: Learning controllable physical priors for generative motion. *arXiv preprint arXiv:2604.28169*, 2026.

William Peebles and Saining Xie. Scalable diffusion models with transformers. In *2023 IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 4172–4182. IEEE, 2023.

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.

Nikhila Ravi, Valentin Gabeur, Yuan-Ting Hu, Ronghang Hu, Chaitanya Ryali, Tengyu Ma, Haitham Khedr, Roman Rädle, Chloe Rolland, Laura Gustafson, et al. Sam 2: Segment anything in images and videos. In *International Conference on Learning Representations*, volume 2025, pp. 28085–28128, 2025.

Team Seedance, De Chen, Liyang Chen, Xin Chen, Ying Chen, Zhuo Chen, Zhuowei Chen, Feng Cheng, Tianheng Cheng, Yufeng Cheng, et al. Seedance 2.0: Advancing video generation for world complexity. *arXiv preprint arXiv:2604.14148*, 2026.

Ying Shen, Jerry Xiong, Tianjiao Yu, and Ismini Lourentzou. Phantom: Physics-infused video generation via joint modeling of visual and latent physical dynamics. *arXiv preprint arXiv:2604.08503*, 2026.

Team Wan, Ang Wang, Baole Ai, Bin Wen, Chaojie Mao, Chen-Wei Xie, Di Chen, Feiwu Yu, Haiming Zhao, Jianxiao Yang, et al. Wan: Open and advanced large-scale video generative models. *arXiv preprint arXiv:2503.20314*, 2025.

Varun Varma Thozhiyoor, Shivam Tripathi, Venkatesh Babu Radhakrishnan, and Anand Bhattad. Principia: Relational physics tests for video models. *arXiv preprint arXiv:2609.04200*, 2026.

Emanuel Todorov, Tom Erez, and Yuval Tassa. Mujoco: A physics engine for model-based control. In *2012 IEEE/RSJ international conference on intelligent robots and systems*, pp. 5026–5033. IEEE, 2012.

Antonios Tragoudaras, Chenyu Zhang, Daniil Cherniavskii, Antonios Vozikis, Thijmen Nijdam, Derck WE Prinzhorn, Mark Bodracska, Nicu Sebe, Andrii Zadaianchuk, and Efstratios Gavves. Evaluating newtonian mechanics in video generative models with real physical systems. *arXiv preprint arXiv:2504.02918*, 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. In D. Belgrave, C. Zhang, H. Lin, R. Pascanu, P. Koniusz, M. Ghassemi, and N. Chen (eds.), *Advances in Neural Information Processing Systems*, volume 38, Main Conference, pp. 167907–167932. Curran Associates, Inc., 2025. doi: 10.52202/085713-5598. URL https://proceedings.neurips.cc/paper_files/paper/2025/file/f53fd88a4340063ecd258c0ae9948b40-Paper-Conference.pdf.

Shuai Wang, Yaxin Feng, Xuekun Jiang, Shihan Tian, Ningyu Yan, Xing Shen, Chaoyang Lyu, Hui Wang, Yunsong Zhou, Hanqing Wang, et al. Gauge: A measurement-grounded benchmark for physical fidelity in simulation engines and video world models. *arXiv preprint arXiv:2608.05948*, 2026.

Haoning Wu, Chaofeng Chen, Liang Liao, Jingwen Hou, Wenxiu Sun, Qiong Yan, Jinwei Gu, and Weisi Lin. Neighbourhood representative sampling for efficient end-to-end video quality assessment. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 45(12):15185–15202, 2023.

Weijia Wu, Zhuang Li, Yuchao Gu, Rui Zhao, Yefei He, David Junhao Zhang, Mike Zheng Shou, Yan Li, Tingting Gao, and Di Zhang. Draganything: Motion control for anything using entity representation. In *European Conference on Computer Vision*, pp. 331–348. Springer, 2024.

Tianyi Xie, Yiwei Zhao, Ying Jiang, and Chenfanfu Jiang. Physanimator: Physics-guided generative cartoon animation. In *2025 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 10793–10804. IEEE, 2025.

Zhuoyi Yang, Jiayan Teng, Wendi Zheng, Ming Ding, Shiyu Huang, Jiazheng Xu, Yuanming Yang, Wenyi Hong, Xiaohan Zhang, Guanyu Feng, et al. Cogvideox: Text-to-video diffusion models with an expert transformer. In *International Conference on Learning Representations*, volume 2025, pp. 83048–83077, 2025.

Lvmin Zhang, Anyi Rao, and Maneesh Agrawala. Adding conditional control to text-to-image diffusion models. In *2023 IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 3813–3824. IEEE, 2023.

APPENDIX CONTENTS

A More Qualitative Results **15**
Additional EnactPhys Results 16

B VLM-Based Physical Plausibility Scoring **17**

C PhysDelta Evaluation Protocol **18**
C.1 PhysDelta-Sim 18
C.2 PhysDelta-Real 19
C.3 Scene-Specific Measurements 20

D Baseline Configurations and Comparison Protocol **21**
D.1 Task Taxonomy and Input Representation 21
D.2 External Benchmark Protocols 21

E Architecture, Training, and Inference **22**
E.1 Reader and Cross-Block State Update 22
E.2 Writer and Spatial Supervision 23
E.3 Optimization and Generation 23

F Training Data and Sample Representation **24**

G Object-State and Interaction Analyses **25**

H Human Verification of Control Scores **26**

I Limitations and Outlook **26**

A MORE QUALITATIVE RESULTS

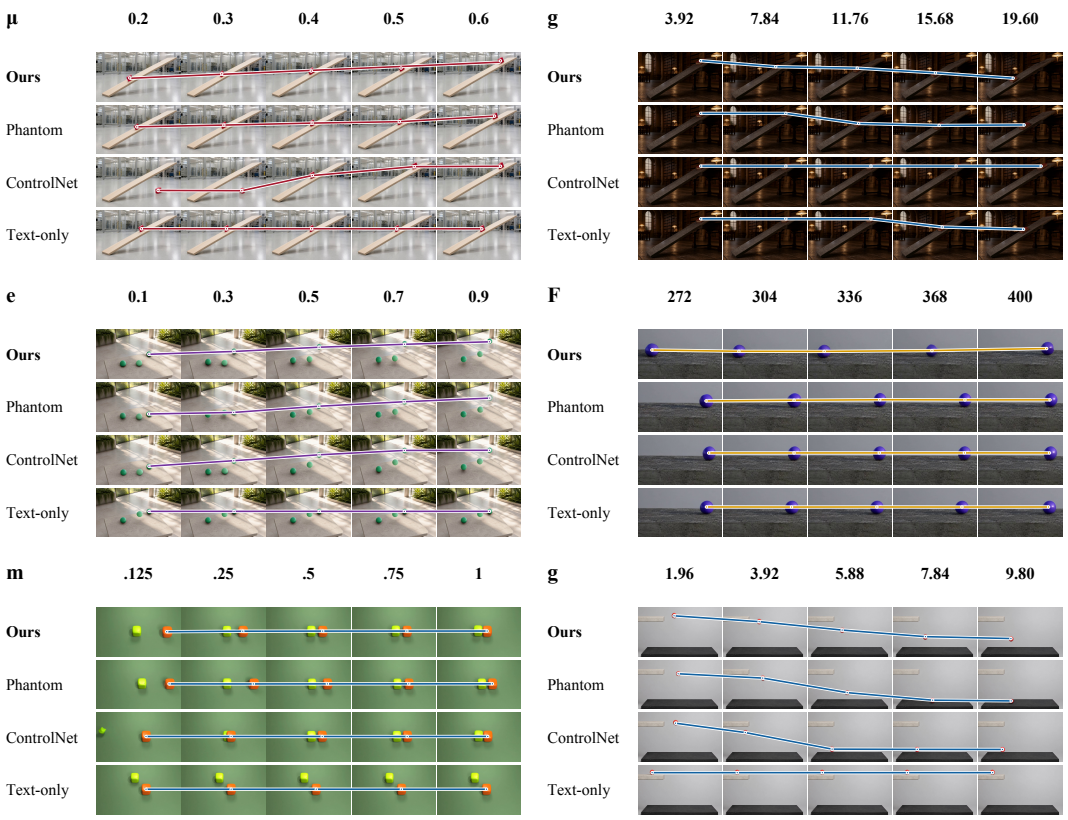

Figure 8: **Five-level physical-control comparisons with baselines using the same Wan2.2-TI2V-5B backbone.** Within each panel, all methods share the initial scene, parameter settings, seed, and observation time. Dots mark observed object positions, and connecting lines visualize responses across parameter levels rather than temporal trajectories.

Case analysis. EnactPhys exhibits **smooth, graded responses** to changes in physical parameters, translating numerical interventions into ordered changes in motion. In contrast, Phantom's responses often **cluster together**: in the F example, four requested settings produce nearly identical positions, while neighboring g settings also yield plateaus rather than consistently graded responses. ControlNet and Text-only exhibit additional failures, including **weak or absent parameter responses** and **incorrect object interactions**. In the collision example, these failures appear as motion inconsistent with the intended impact or a failure to establish contact; in the bottom-right g example, ControlNet's three highest settings converge to similar positions, while Text-only remains on the ledge across all five settings. These qualitative observations are consistent with EnactPhys's higher parameter-control and object-control scores in Table 2.

ADDITIONAL ENACTPHYS RESULTS

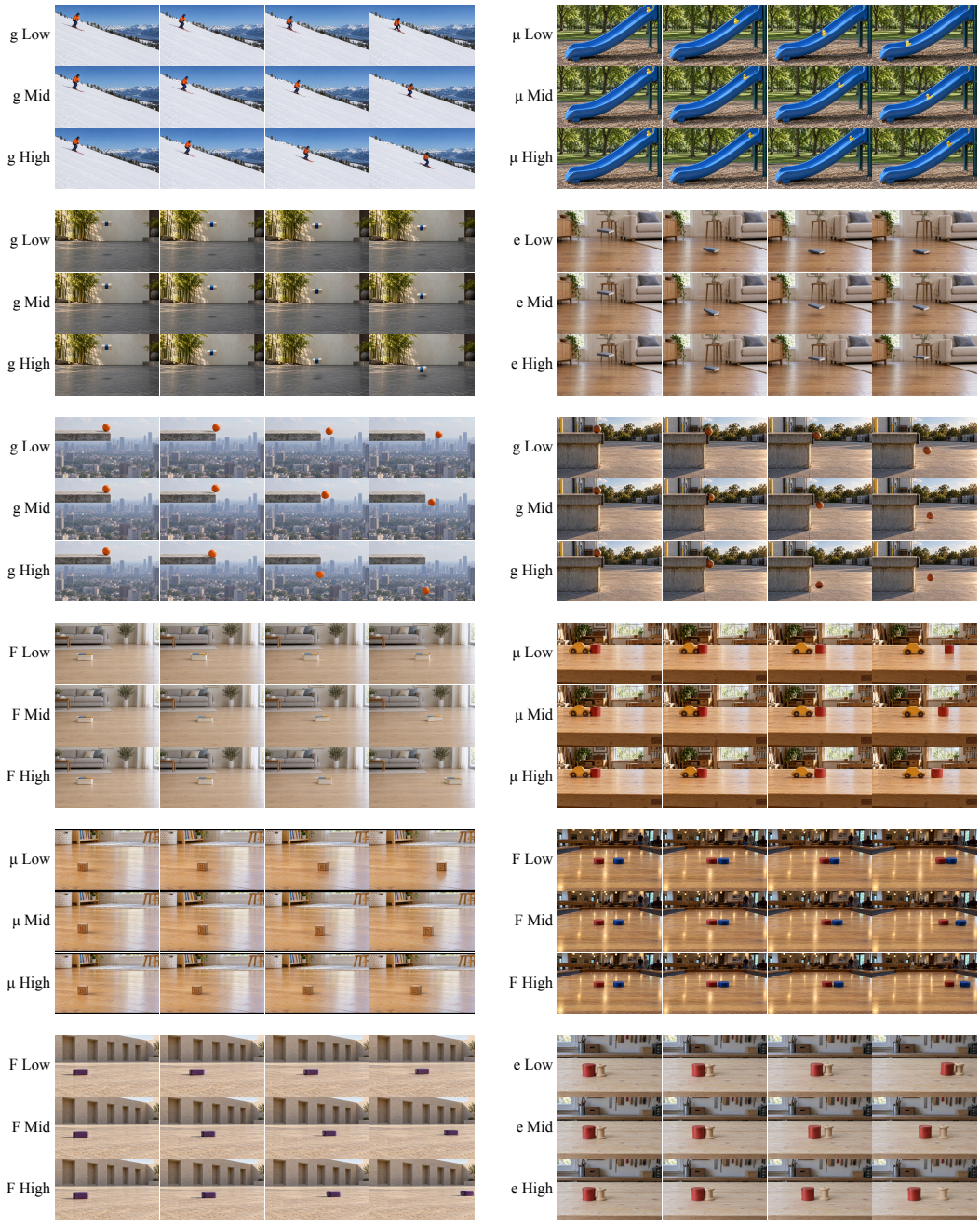

Figure 9: **Additional EnactPhys results.** Twelve examples of physical control across varied scenes. Within each example, rows vary the indicated physical quantity from low to high, and columns show successive frames at matched times. The fixed initial scene makes the resulting differences in motion visible across the requested settings.

B VLM-BASED PHYSICAL PLAUSIBILITY SCORING

Physical Plausibility (PP) measures visible generation abnormalities. We uniformly sample 16 frames from each video and arrange them in a chronological 4×4 grid. Qwen3-VL-235B-A22B-Instruct (Bai et al., 2025) evaluates the grid with temperature 0, without the generation prompt or requested physical values. It returns a judgment, an issue category, and brief visual evidence. All methods use the same sampling, instructions, and evaluator settings. We will release the full prompt and inference configuration with the evaluation code.

Scoring. A valid evaluator response is converted to a binary decision using its label and issue category. A video fails if the evaluator explicitly flags one of the listed abnormalities other than disappearance, or if SAM 2 (Ravi et al., 2025) trajectories indicate an in-view disappearance rather than an exit at the image boundary. Otherwise, it passes. Valid `UNCERTAIN` responses remain in the denominator and count as passes unless the trajectories indicate an in-view disappearance. PP is the proportion of videos that pass; PC and OC separately evaluate parameter-dependent response and object selectivity.

API failures and malformed responses are unresolved evaluations, not pass/fail judgments. These cases are retried or resolved by human review under the same criteria before final main-table aggregation, which uses the full predefined sample denominator. The evaluator–human agreement analysis below includes only valid automated evaluations.

Physical Plausibility evaluator prompt (verbatim).

```
The image shows sixteen complete video frames sampled uniformly in
time and arranged as a 4x4 grid.  Read left-to-right, top-to-bottom,
t01 to t16.  Decide only whether there is a CLEAR visual generation
abnormality.

FAIL for direct visible evidence of:  a large colorful wave/ring or
abstract structure; unexplained particles, lines, shards or debris;
human/hand/body intrusion; a coherent new object appearing from
nowhere; two complete copies of one target simultaneously visible in
the same tile; a rigid target persistently breaking, melting,
shrinking, stretching or deforming; persistent
identity/color/material mutation; persistent background/support
mutation or a global camera jump; an abnormal translucent afterimage
within one tile; a true in-view disappearance; or unmistakable
rigid-body penetration.

PASS when no such defect is directly visible.  Different tiles are
different times, so different positions across tiles are never
duplication.  Because the frames are sparse samples, later absence
alone is not proof of disappearance:  normal motion toward and
beyond a local image border is PASS even if the exact crossing
instant was not sampled.  Contact, resting on a surface, partial
occlusion by an edge, perspective overlap, and one ambiguous overlap
are not penetration.  Normal motion, collision, bounce, rotation,
lighting, shadow, reflection, compression and ordinary motion blur
are PASS. Do not judge force magnitude, friction, gravity,
restitution, speed, trajectory, timing or control accuracy.

Choose the single most decisive issue_type from:  wave_or_abstract,
particles_or_debris, human_intrusion, new_object, duplication,
deformation, identity_change, scene_or_camera_change,
abnormal_afterimage, disappearance, penetration, none, uncertain.
Use FAIL only for decisive visible evidence; otherwise PASS.

Return JSON only:  {"label":"PASS|FAIL|UNCERTAIN","issue_type":"one
value from the list","evidence":"one short sentence naming the
visible evidence and tile numbers"}
```

Agreement with human judgments. A human reviewer watched the complete videos and assigned pass/fail judgments under the same plausibility criteria. Of 205 human-reviewed videos, 204 have valid automated evaluations; one video with an unresolved API failure is excluded from this agreement analysis. The evaluator agrees with the human judgments on 187 of the 204 videos (91.67%), with Cohen's $\kappa = 0.8082$.

C PHYSDELTA EVALUATION PROTOCOL

PhysDelta evaluates the effect of changing a physical condition while holding the initial image, object assignment, remaining conditions, and random seed fixed. PhysDelta-Sim measures the direction and magnitude of physical responses, their invariance where appropriate, and object selectivity. PhysDelta-Real measures response ordering and object selectivity in photographic scenes. The metric definitions in Section 4.3 are instantiated below through scene-specific motion measurements and matched comparisons. The scene overview in Figure 3 lists the physical quantities involved in each family. In the tables below, "Scanned quantities" and intervention axes identify the quantities varied in the specified tests; within each single-axis comparison, the other physical conditions remain fixed.

C.1 PHYSDELTA-SIM

The simulated track covers sliding, falling, collisions, motion on inclines, projectile motion, and balance. Table 4 identifies the physical event and expected response for each family; Section C.3 describes trajectory measurement and score aggregation.

Table 4: PhysDelta-Sim response measurements. Each intervention fixes the initial image and other conditions. m_d and m_r denote driver and receiver mass. Measurement and pair-score definitions are given in Section C.3.

Scene	Intervention	Response and observation phase
S01	F, m, μ	F, m: maximum travel along the force direction over the valid track; μ: free deceleration after the force interval.
S02	g, e	g: downward motion during descent; e: first rebound height, normalized by initial object height.
S03	e	Outgoing-to-incoming speed ratio along the approach direction at the first wall impact.
S04	F, m, g	F, m: horizontal travel during the initial airborne phase; g: downward motion after support departure.
S05	g, μ	Along-ramp motion while supported; g: gravity response; μ: decreasing fitted speed.
S06	F, μ	F: driver approach speed before contact; μ: deceleration in an isolated post-force segment.
	e, m_r, m_d	e, m_r: receiver normal-speed gain at the first collision; m_d: receiver gain relative to incoming relative speed.
S07	g, μ	Supported downhill response before ramp departure; μ: decreasing fitted downhill speed.
S08	F, μ, g	F: post-force speed; μ: free deceleration; g: downward motion after platform departure.
S09	g, μ, e	g, μ: supported ramp motion; e: ordering of the normal rebound response at the wall.
S10	F, g, e	F: post-force speed; g: airborne downward motion; e: outgoing-to-incoming speed ratio at ground impact.
S11	g, e	g: downward motion before impact; e: vertical first-rebound height normalized by initial object height.
S12	g, μ	g: maximum travel projected along the ramp direction; μ: decreasing fitted speed while on the ramp.
B	m_L, m_R	Change in left-minus-right vertical position during the first 0.5 s; its sign is compared with $m_L - m_R$.

PhysDelta-Sim covers 13 scene families. Its core Parameter Control set contains 71 initial-image cases and 2,385 generated videos: 795 condition states evaluated with each of three seeds. Sixty additional videos evaluate invariant quantities, and the Object Control track contains 201 videos, giving 2,646 videos per method in total. Table 5 lists the unit of evaluation for every scene family. The "videos" column is the core Parameter Control count across three seeds; Object Control and invariance tasks are described separately below.

Table 5: PhysDelta-Sim task families and evaluation volume. Scene-level readouts are specified in Table 4.

ID	Scene family	Cases	Videos	Scanned quantities
S01	Forced Sliding	5	195	F, m, μ
S02	Drop & Bounce	5	135	g, e
S03	Wall Impact	5	75	e
S04	Projectile Motion	5	195	F, g, m
S05	Ramp Sliding	7	189	g, μ
S06	Object Collision	9	471	F, e, m_d, m_r, μ
S07	Ramp-to-Flat	5	135	g, μ
S08	Platform Exit	5	195	F, g, μ
S09	Ramp-Wall Impact	5	195	g, e, μ
S10	Projectile Bounce	5	195	F, g, e
S11	Incline Bounce	5	135	g, e
S12	Ramp Exit	5	135	g, μ
B	Beam Balance	5	135	m_L, m_R
	Total	71	2,385	F, g, m, μ, e

The five-level sweeps preserve the initial image, target assignment, all non-scanned physical values, and seed. Each scanned axis has a center condition and two lower and two higher levels; the center condition is shared across axes within a case. The quantities listed in Table 5 denote the union of axes scanned within each scene family. In S06, all nine cases scan F, μ, and e, and five of these also scan m_d and m_r. Thus, five cases contain 21 condition states and four contain 13, giving $(5 \times 21 + 4 \times 13) \times 3 = 471$ videos across three seeds. The invariance set measures motion quantities that should remain unchanged under the specified intervention and includes added mass interventions in S02 and S05. The Sim Object Control set contains 67 condition states per seed over three seeds. It intervenes on F, g, or e for one designated object while retaining the conditions of the remaining objects.

C.2 PHYSDELTA-REAL

PhysDelta-Real covers 12 scene families, corresponding to S01–S12 in PhysDelta-Sim. Its Parameter Control track defines 268 condition inputs per seed from clean pre-event frames, giving 804 videos over three seeds. Later frames from the source videos are not used as generation targets. Table 6 lists the conditions and the exact single-axis comparison denominator for each scene. The readout is the largest centroid displacement from its initial position: horizontal for S01, S03, S04, S06, S08, and S10; downward for S02 and S11; and Euclidean for S05, S07, S09, and S12. S09 supplies joint-condition examples but no single-axis scoring pair.

Table 6: PhysDelta-Real Parameter Control scene protocol. "Conditions" counts generation inputs per seed; "pairs" counts fixed scored comparisons per seed. Higher F and g must increase the listed response, whereas higher μ must decrease it.

ID	Scene family	Conditions	Axis	Pairs	Video readout
S01	Forced Sliding	26	F	13	horizontal displacement
S02	Drop & Bounce	27	g	27	downward displacement
S03	Wall Impact	21	F	12	horizontal displacement
S04	Projectile Motion	14	F	7	horizontal displacement
S05	Ramp Sliding	64	g, μ	98	two-dimensional displacement; matched single-axis pairs
S06	Object Collision	36	F	36	driver and receiver horizontal displacements
S07	Ramp-to-Flat	18	μ	18	two-dimensional displacement
S08	Platform Exit	12	F	6	horizontal displacement
S09	Ramp-Wall Impact	12	g, μ	0	joint-condition examples
S10	Projectile Bounce	16	F	8	horizontal displacement
S11	Incline Bounce	16	g	8	downward displacement
S12	Ramp Exit	6	g	3	two-dimensional displacement
	Total	268		236	

The Real Object Control track uses five real initial scenes: four multi-object drop scenes for g and e, and one three-object sliding scene for F. It contains 69 generated videos per method over three fixed seeds. Object Control uses 6 force comparisons, 24 gravity comparisons, and 24 restitution comparisons per method.

C.3 Scene-Specific Measurements

Section 4.3 defines PC and OC. Here we specify the motion measurements used for each task. SAM 2 (Ravi et al., 2025) tracks objects from their first-frame masks; mask centroids provide image-plane trajectories. Coordinates are mapped to a common 768×448 canvas, and observation intervals are expressed in seconds.

Parameter Control on PhysDelta-Sim. Table 4 specifies the response and motion phase for each scene. Each five-level sweep yields ten comparisons with the initial image, non-intervened conditions, and seed fixed. The direction and magnitude components average comparisons within object, seed, case, and scene–parameter group, with equal weight for F, g, m, μ, e. The final PC score averages the magnitude, direction, and invariance components, weighted by their respective numbers of test groups. Missing required motion events and unreadable responses contribute zero; the comparison set is fixed across methods.

Parameter Control on PhysDelta-Real. Table 6 specifies the displacement readout for each scene. Higher force and gravity should increase the response, while higher friction should reduce it. A comparison succeeds when the response changes in the expected direction by more than 0.5 pixels. Scores average the fixed 708 comparisons across three seeds; unreadable responses contribute zero.

Object Control. Each intervention is compared with its center condition under the same initial image and seed. Motion differences are average distances between matched trajectories. Non-target change is the mean across other objects for force, and the maximum for gravity and restitution.

Table 7: Object Control response measurements. Each domain applies the same task protocol to all compared models.

Quantity	Target response	Observation interval
F	displacement along the applied force	first 0.5 s, before inter-object contact
g	pre-impact acceleration	up to 0.5 s, before support or inter-object contact
e (Sim)	outgoing-to-incoming normal speed ratio	first support contact; trajectory comparison extends five frames beyond the later contact, up to 2 s
e (Real)	rebound height from support contact to the post-contact apex	first 2 s

The response factor c uses directional success for force. For gravity and Sim restitution, it measures the fraction of the expected response change attained in the requested direction, capped at one. For Real restitution, it requires a rebound-height change greater than 5 pixels in the requested direction. Target trajectory differences below the repeat-derived tracking threshold receive zero. The final OC weights F, g, e equally: Sim uses 48, 60, and 60 comparisons, and Real uses 6, 24, and 24, respectively. Human verification of these assessments is reported in Section H.

Video Quality and Physical Plausibility. Sim VQ and PP each use a fixed panel of 215 videos per method: 200 PC videos and 15 OC videos. The two panels share 202 videos and differ in 13 Forced Sliding videos. For these 13 entries, the PP panel uses matched replacements generated from validation-set initial frames with the same parameter settings and seeds; the VQ panel retains the original videos. Each metric uses the same task selection across methods.

Real VQ and PP use the same 100 videos per method: 75 additional PC videos generated with seed 42 and 25 OC videos already included in the 69-video OC set. Thus, PhysDelta-Real contains 948 distinct videos per method ($804 + 69 + 75$), and PhysDelta contains 3,594 across Sim and Real. PP follows Section B; the Physics-IQ protocol is given in Section D.

D BASELINE CONFIGURATIONS AND COMPARISON PROTOCOL

All methods receive the same PhysDelta task definitions, initial images, intervention values, target-object assignments, and paired seeds. The three trained Wan2.2 comparisons—Wan (Text-only), Wan (ControlNet), and Phantom—use the same Wan2.2-TI2V-5B initialization, 109,800 training videos, 6,000-step budget, output length, and spatial resolution as EnactPhys. Their condition representations differ, while the training clips, physical values, and evaluation tasks remain matched.

General-purpose image-to-video models receive object-specific physical conditions through structured text. Force Prompting and PhyCo receive conditions through their native control interfaces where supported; the remaining conditions are conveyed through the same structured-text format. For each PhysDelta metric, all methods use the same evaluation set, measurement procedure, scorer, and aggregation denominator.

D.1 TASK TAXONOMY AND INPUT REPRESENTATION

A task combines an initial image, object masks, and physical conditions. Force, gravity, and mass are bound to individual objects; friction and restitution describe object–object or object–support relations. Figure 10 illustrates these bindings in three scene families. A force intervention changes the designated object's force while holding other conditions fixed; a collision intervention changes a mass or contact property while preserving object identities and relation endpoints.

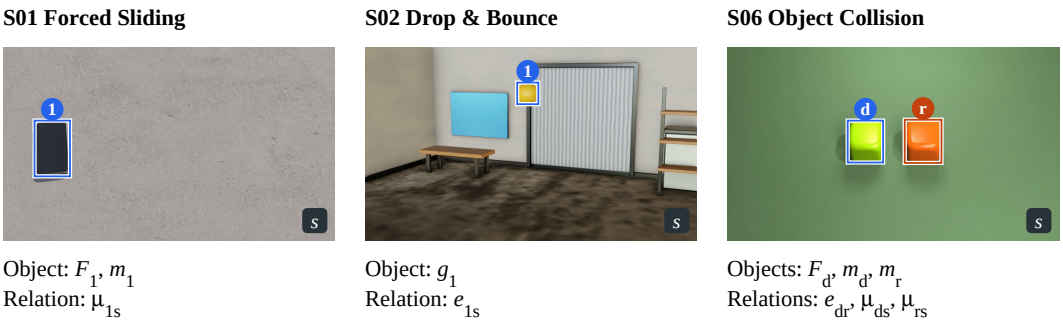

Figure 10: **Examples of physical-condition inputs.** Initial images from Forced Sliding, Drop & Bounce, and Object Collision. Boxes and labels identify the objects using their first-frame masks; s denotes the support. Labels below each image show the associated condition fields, with d and r marking the collision driver and receiver.

D.2 EXTERNAL BENCHMARK PROTOCOLS

Physical inputs. We retain the official scene descriptions and supply physical conditions through the supported control interfaces. EnactPhys uses first-frame masks to bind conditions to objects and support regions. Per-case inputs and branch settings are specified in the evaluation configuration and held fixed across seeds. Where physical conditioning is not used, the entire physical-conditioning branch is disabled.

Physics-IQ Verified. We evaluate all 38 Solid Mechanics scenes with three views and three seeds, yielding 342 results per method (Motamed et al., 2026; Rädsch et al., 2026). The matched Wan-based controlled comparisons share the configured physical inputs; PhyCo retains its native per-case conditions. Following PhyCo's training-length evaluation setting (Narayanan et al., 2026), EnactPhys's 49-frame, 24-FPS outputs are extended to 120 frames by repeating the final frame; PhyCo uses its 57-frame setting with the same padding procedure. All methods use the official Verified/View scorer and reference data, with equal weighting across views and seeds.

MORPHEUS (adapted protocol). We evaluate 16 experiments with three seeds, yielding 48 results per method (Tragoudaras et al., 2025). Videos are evaluated over a common 49-frame interval at 24 FPS; longer outputs are truncated, and native 8-FPS outputs are resampled. We retain the released Dynamical and Physical Invariance scoring formulas, with a shared short-video adaptation to falling-trajectory truncation and visual validity screening. Confirmed invalid videos receive zero and remain in the denominator; other cases retain their trajectory scores. All methods use the same

adapted protocol, with equal weighting across experiments and seeds. Exact input and temporal conversion settings are provided with the evaluation code.

E ARCHITECTURE, TRAINING, AND INFERENCE

EnactPhys augments the frozen Wan2.2-TI2V-5B backbone with eight object-state modules, inserted within zero-indexed backbone blocks 10–17, after self-attention and before cross-attention. Each module performs Reader, unary conditioning, same-object temporal attention, pairwise interaction, and Writer updates in that order. Modules have separate parameters and exchange object states and predicted Reader routes within each denoising pass.

Table 8: Dimensions and routing configuration of the object-state modules.

Component	**Configuration**
Video / latent times	49 frames / 13 latent positions
Object-state width	512
Spatial query/key width	128
Temporal attention	8 heads; same object across all latent times
Spatial routing grid	14×24
Reader support	top 8 locations per object and latent time
Writer support	top 8 locations in a radius-1 square
Reader route prior	previous module's predicted route; weight 1.0
Support representation	one static support entity for interactions

E.1 READER AND CROSS-BLOCK STATE UPDATE

The Reader connects each object's identity to the evolving video features. The first-frame mask supplies an identity anchor, while content-based attention locates relevant features separately at each latent time. Thus, an object's read locations can move with the generated motion while its state sequence retains the same object index.

Let $X_{t,p}$ denote a video token at latent time t and spatial cell p, and let $M_o(p)$ be object o's first-frame mask on the routing grid. The Reader anchors identity by pooling normalized first-frame features:

$$a_o = W_v \left(\frac{\sum_p M_o(p)\, \mathrm{LN}(X_{0,p})}{\sum_p M_o(p)} \right) + \phi(\mathrm{geom}(M_o)), \qquad u_{o,t} = a_o + \tau_t. \tag{5}$$

The five geometry features are the mask's two center coordinates, two spatial spreads, and area fraction. τ_t is a learned temporal embedding. Queries $q_{o,t} = W_q\, \mathrm{LN}(u_{o,t})$ and keys $k_{t,p} = W_k\, \mathrm{LN}(X_{t,p})$ have width 128.

Spatial readout. The Reader scores locations by scaled query–key products. A predicted route from the previous module contributes an additive spatial prior:

$$\ell_{o,t,p} = q_{o,t}^{\mathsf{T}} k_{t,p} / \sqrt{128} + P_{o,t,p}, \qquad R_{o,t} = u_{o,t} + W_r \sum_{p \in \mathcal{K}_{o,t}} \alpha_{o,t,p} W_v\, \mathrm{LN}(X_{t,p}). \tag{6}$$

$\mathcal{K}_{o,t}$ contains the top 8 locations and α is the softmax over their scores. The prior is formed by scattering the previous Reader weights onto the grid and applying 3×3 max pooling; its coefficient is 1.0. The first module has no preceding route. The routing and video-token grids both have size 14×24. Because the prior is added to the current content scores, subsequent modules can revise the selected locations. The weighted read aggregates visual evidence for one object at one latent time; its residual connection retains the identity and temporal embedding.

State fusion. At the first module, R initializes the object states. At subsequent modules, the current read is fused with the preceding module's output state S^-:

$$\gamma_{o,t} = \sigma\big(W_f[\mathrm{LN}(S^-_{o,t}); \mathrm{LN}(R_{o,t})] + b_f\big)\,, \tag{7}$$

$$S^{\mathrm{read}}_{o,t} = \gamma_{o,t} \odot S^-_{o,t} + (1 - \gamma_{o,t}) \odot R_{o,t}. \tag{8}$$

The gate has one value per state channel, allowing each channel to retain information from the preceding state or incorporate the current visual read. This fusion carries the preceding module's physical updates into the next module while refreshing the state with its current video features. Object states and routes are carried through the eight modules of one denoising pass and initialized again for the next pass. This is distinct from the latent-time axis over which each object's temporal attention operates.

Physical updates and interactions. Force, gravity, and mass embeddings are added to S^{read}, with force following its active-time schedule. Temporal attention exchanges information across the 13 times. Pairwise updates then combine base, restitution, and friction messages under separate contact gates. The resulting S^{out} passes to the Writer and next module; support participates only in interactions.

E.2 Writer and Spatial Supervision

The Writer returns the physically conditioned object states to the video representation. It separates the content to be written, obtained by projecting each updated state, from its spatial placement, determined by object-specific attention weights at each latent time.

Local write support. The Writer has its own query and key projections. Its query comes from the updated state $S^{\mathrm{out}}_{o,t}$, and its keys come from the current video tokens. The current Reader weights define a spatial anchor by their weighted mean grid coordinate, rounded to the nearest cell. The Writer selects up to 8 locations inside the radius-1 square around this anchor and computes a softmax over its own scores on these locations. Reader and Writer thus share object localization information while learning separate read and write weights. The Reader anchor identifies the neighborhood in which to write; the updated state determines how the residual is distributed within that neighborhood. This places the state update near the object's predicted location even when that location differs from its position in the first image.

Residual writeback. Let $\beta_{o,t,p}$ denote the sparse Writer weights and $V_{o,t} = W_o S^{\mathrm{out}}_{o,t}$ the projected state. Writeback is

$$\Delta X_{t,p} = \frac{\sum_o \beta_{o,t,p} V_{o,t}}{\max(1, \sum_o \beta_{o,t,p})}, \qquad X'_{t,p} = X_{t,p} + \eta(d)\Delta X_{t,p}, \quad t > 0. \tag{9}$$

The denominator bounds the total incoming weight when object supports overlap. $\eta(d)$ combines the learned residual scale and a sigmoid gate conditioned on diffusion timestep d. The first latent position receives zero write residual, preserving the image-conditioning tokens. Outside the selected Writer locations, this module contributes no residual. The modified video features then pass through the following backbone blocks, where the next Reader can gather evidence from the updated representation.

Spatial supervision. Per-frame masks provide training targets at the latent positions. For either Reader or Writer, let Y_p be the object mask on the routing grid, $\widehat{Y}_p = Y_p / \sum_p Y_p$, ℓ_p the spatial logits, and w_p the actual sparse routing weights. Each localization objective combines

$$\begin{aligned} L_{\mathrm{route}} &= -\frac{\sum_p \widehat{Y}_p \log \mathrm{softmax}(\ell)_p}{\log(14 \cdot 24)}, \quad L_{\mathrm{off}} = 1 - \sum_p w_p \mathbf{1}[Y_p \geq 0.1], \\ L_{\mathrm{loc}} &= \tfrac{1}{2}(L_{\mathrm{route}} + L_{\mathrm{off}}). \end{aligned} \tag{10}$$

Losses average over modules and valid object–time entries after the first latent position, omitting entries with empty target masks. Contact labels supervise the interaction gates. These labels enter the losses, not the generation forward pass. At inference, first-frame masks identify objects; subsequent states and routes are predicted from video features and conditions.

E.3 Optimization and Generation

The video backbone remains frozen while the object-state modules are trained for 6,000 steps. The video objective adds temporal-difference and motion-weighted losses, each weighted by 0.2, to the

Table 9: Training and generation settings for EnactPhys.

Setting	Value
Training / validation videos	109,800 / 4,196
Optimizer	AdamW
Learning rate / weight decay	10^{-4} / 0.01
Training steps	6,000
Devices / per-device batch	16 NVIDIA A100 80GB / 2 videos
Global batch / gradient accumulation	32 / 1
Clip shape	49 frames at 768×448
Denoising steps / guidance	30 / 1.2

flow-matching loss. Reader and Writer localization terms each have weight 0.3; predicted contact gates are supervised by binary cross-entropy. The complete objective is given in Eq. 4.

Each PhysDelta comparison holds the initial image, object identities, non-intervened conditions, and random seed fixed. Tracked coordinates are mapped to the common 768×448 canvas for measurement. Video-duration handling is specified in the Physics-IQ Verified and MORPHEUS protocols in Section D.

F Training Data and Sample Representation

The physical-control corpus contains 109,800 training videos and 4,196 validation videos. As described in Section 3.2, the data combine MuJoCo rigid-body simulation with Blender Eevee rendering, a subset of PhyCo data, and samples generated with the PhyCo simulation pipeline. Per-frame object masks and contact labels provide supervision for spatial localization and interaction modeling.

Corpus composition. Table 10 groups the training videos by their dominant motion event. Collision, drop, sliding, ramp, projectile, balance, and partial-occlusion examples jointly expose the model to free motion, support contact, object–object contact, and transitions between these phases. Parameter values are varied both individually and jointly; a single video may therefore provide supervision for more than one physical quantity.

Table 10: Composition of the 109,800-video training corpus. Categories follow the dominant motion event represented by each sample.

Motion group	Videos	Main supervision provided
Object–object collision	29,534	force, mass, restitution, and interaction
Vertical drop and bounce	25,473	gravity, restitution, and support contact
Force–friction sliding	14,271	force, mass, friction, and free deceleration
Inclined-surface motion	14,234	gravity, friction, and support relation
Object–wall collision	10,243	incident motion and restitution
Projectile and edge departure	9,000	force, mass, gravity, and phase transition
Balance scale	4,800	relative mass and multi-object response
Partial occlusion	2,245	object identity and spatial persistence
Total	109,800	

Sample representation. Each video has an ordered set of object slots and one support slot. First-frame masks bind physical conditions to objects; per-frame masks supervise Reader and Writer localization, and contact labels supervise interaction gates. A 49-frame clip uses 13 latent positions on a 14×24 spatial grid. At inference, object identities are initialized from the first-frame masks, with subsequent states and routes predicted from video features and physical conditions.

G OBJECT-STATE AND INTERACTION ANALYSES

These analyses examine whether object states represent physical motion and whether parameter-dependent changes in those states affect generated videos. The state readout and affine transformation evaluate representation and transfer across inputs; contact-window and message interventions examine when and through which interactions physical conditions influence motion.

State-speed readout. Object speed is computed from SAM 2 (Ravi et al., 2025) centroid tracks. For each of the 12 motion slots, a linear fit to the corresponding four video frames estimates horizontal and vertical velocity; their Euclidean norm is the speed target in pixels per frame. The probe reads the 512-dimensional object state after interaction at block 17 of the final conditional denoising pass.

The probe uses a 42/12/12 split of initial-image inputs for fitting, validation, and testing. Inputs and targets are standardized using the fitting split; ridge regularization is selected on the validation split, and the fitted probe is kept fixed for testing. The test comprises 12 held-out inputs, three physical settings per input, and ten seeds, yielding 360 videos and 5,238 valid object–time measurements. Across these points, the probe achieves $R^2 = 0.811$ and a mean absolute error of 1.48 pixels per frame. Figure 5 compares its predictions with tracked speed over time.

Affine state transformation. For a matched pair of physical values, states are aligned by object identity and latent time. An affine map is estimated from the first value to the second on one set of inputs and then applied to different inputs. State prediction and the resulting change in generated motion are evaluated separately: the former measures representational consistency, while the latter tests whether the mapped state affects the video output. Relative to leaving states unchanged, the held-out state-prediction mean squared error decreases by 79.9%. Using Parameter intervention as the reference, applying the map during generation reduces trajectory mean squared error by 94.6% relative to Normal generation. On the evaluated inputs, these results link the transferred state change to the corresponding change in generated motion.

Contact-window intervention. Each case uses one detected contact interval for all compared variants. The base and full-intervention videos hold the initial image, seed, and other conditions fixed. The windowed variant applies the changed pairwise condition only inside that interval and retains the base condition elsewhere. Motion is measured in the post-contact phase relevant to the scene. In the evaluated contact scenarios, intervening only within the contact interval produces motion close to that of the full intervention. This indicates that the parameter-dependent response in these scenarios is concentrated within the contact interval.

Interaction-message intervention. The requested physical conditions remain present while the output of the corresponding restitution or friction message branch is set to zero. All other state updates and the video backbone remain active. For each parameter sweep, motion changes are measured relative to the lowest parameter setting. The response curves become substantially flatter when the corresponding messages are removed. This identifies the typed interaction messages as a pathway through which the supplied physical parameters affect generated motion.

H HUMAN VERIFICATION OF CONTROL SCORES

We selected generated-video pairs from PhysDelta-Sim and PhysDelta-Real for human review of Parameter Control and Object Control. A reviewer watched the paired videos alongside their physical conditions, automatic motion readouts, and scores, and judged whether the automatic assessment agreed with the visible motion. Across 249 reviewed pairs, 242 assessments were consistent with human judgments (97.19% agreement; Table 11).

Table 11: Human agreement with automatic control scores. The overall rate pools reviewed pairs across the four metric–domain groups.

Domain	**Metric**	**Reviewed**	**Agreed**	**Agreement**
PhysDelta-Sim	Parameter Control	84	82	97.62%
PhysDelta-Sim	Object Control	81	79	97.53%
PhysDelta-Real	Parameter Control	27	25	92.59%
PhysDelta-Real	Object Control	57	56	98.25%
Overall		**249**	**242**	**97.19%**

I LIMITATIONS AND OUTLOOK

Our experiments focus on object-centric rigid-body dynamics governed by force, gravity, mass, friction, and restitution. Although these dynamics capture a broad range of everyday physical interactions, extending the framework to deformable objects and fluids remains an important direction for future work.

Our evaluation of EnactPhys focuses on short, 49-frame video clips. Extending this approach to long-video and autoregressive generation is a further direction for future work, with an emphasis on maintaining coherent object states and parameter-dependent interactions across successive video segments.