~~STRIKE~~: LEARNING VISUAL STATE TRANSITIONS FOR PHYSICAL WORLD MODELING

Anonymous authors
Paper under double-blind review

ABSTRACT

Physical world modeling requires predicting how interactions change a scene, not merely generating coherent motion. We propose ~~STRIKE~~, a framework that separates visual state transition learning from dense video generation. We construct event-aligned supervision by extracting observed states from training videos and pairing them with transition descriptions and temporal offsets. An image-based transition model learns to predict the next scene configuration from the current image, a local transition specification, and elapsed time. At inference, a pretrained vision-language planner predicts time transition specifications, and recursive application of the learned transition model produces a sequence of future visual states. A separately trained dynamic model then generates the complete rollout conditioned on these states and their temporal locations. Experiments on Physics-IQ Verified, PhyGenBench, Pisa-Experiments, and RoboTwin2.0 show improvements of ~~STRIKE~~ over the corresponding video-backbone baselines in benchmark measures of physical consistency and manipulation-video fidelity. These results support learned visual state transitions as an effective intermediate representation for physical world modeling.

1 INTRODUCTION

A central challenge in physical world modeling is predicting the consequences of interactions among objects and agents. For example, a successful grasp establishes attachment between an object and a gripper, while lift preserves that attachment and changes the object's support conditions (Figure 1). We use visual states to denote configurations that express such interaction outcomes (Zeng et al., 2024). These states capture changes in object configurations and relations that constrain subsequent physical evolution. Preserving object appearance alone is therefore insufficient: the generated scene must also establish the relation of transition that make an interaction physically meaningful.

We consider a formulation that treats transitions between visual states as an explicit learning target. A state-transition model predicts the state reached after an interaction, given the current observation and an interaction specification. A conditional dynamics model then models how the scene evolves between the predicted states. For open-domain generation, a pretrained vision-language event planner provides the local transition specifications and their temporal locations. This formulation allows us to investigate whether explicitly learning state transitions improves the prediction of interaction outcomes and their subsequent physical consequences.

Learning state transitions is a general approach to predictive modeling for planning and control. Latent dynamics models support planning and behavior learning (Hafner et al., 2019; 2020), visual prediction supports robot manipulation (Ebert et al., 2018), and language-conditioned video models represent task execution through visual plans (Du et al., 2023). Within this broader setting, we study a transition interface defined by the current image, a local description of a physical change, and elapsed time. Our focus is on learning this mapping from event-aligned state pairs and composing its predictions to guide dense visual evolution.

In this work, we propose ~~STRIKE~~, a novel framework that explicitly learns visual state transitions and uses the predicted states to condition dense video generation. We first instantiate the state-transition model as a conditional image model that predicts a future visual state from the current observation, an interaction specification, and the elapsed time. To construct its supervision, a VLM-assisted curation pipeline identifies events in training videos, selects informative visual states, and

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

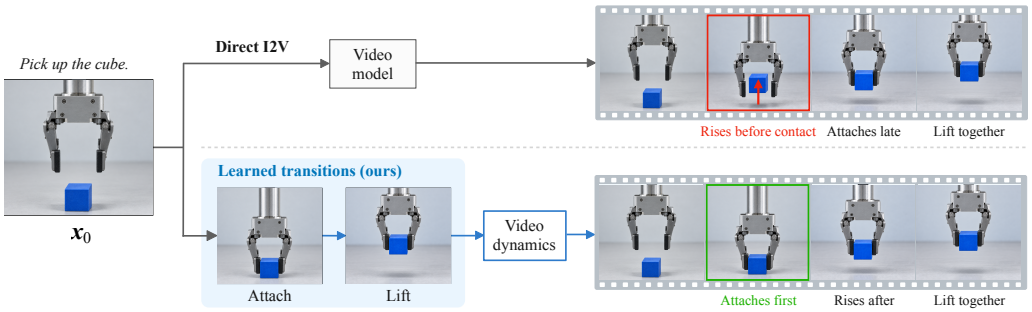


Figure 1: **Overview.** Given an initial image and task, ~~STRIKE~~ uses a timed event plan to recursively predict future visual states with a transition model learned from event-aligned state pairs. The predicted states and their temporal positions are applied as conditions of dense video generation.

annotates the transitions between them. After structural and quality checks, adjacent selected states are paired with transition descriptions and elapsed times, yielding supervision grounded in observed changes rather than generic video-level captions. The selected images are extracted from training videos rather than synthesized by the annotator. We train the image model on these pairs, directly supervising how a specified interaction changes the scene's object configurations and relations. During inference, the event planner instead predicts future transition conditions from the initial image and task context, without observing future reference frames. Each predicted state becomes the input for the next specified interaction, allowing the model to construct a sequence of successive visual states. These predictions provide explicit state and timing conditions for dense video generation. Second, the conditional dynamics model is implemented as a video diffusion model trained to generate the motion and intermediate configurations connecting these states. Together, the two components model physical evolution through explicit interaction outcomes and the dynamics between them.

Our contributions are as follows:

- We formulate visual state-transition prediction as an explicit event- and time-conditioned learning task, and construct event-aligned supervision from observed video states through a VLM-assisted annotation pipeline.
- We develop a compositional generation framework that combines timed event planning with recursive prediction of transition states. These states and their temporal locations provide soft conditions for separately trained video dynamics models, instantiated with CogVideoX and Wan.
- We evaluate the framework on Physics-IQ Verified, PhyGenBench, PisaBench, and RoboTwin2.0, demonstrating gains in physical-consistency and manipulation-video metrics over the corresponding backbone baselines.

2 Related Work

Physics-Aware Generative World Models. Large-scale video generators, including CogVideoX (Yang et al., 2025b), Wan (Wan et al., 2025), and Cosmos (Ali et al., 2025; NVIDIA, 2026), provide backbones for visual world modeling. IRASim (Zhu et al., 2025b) aligns action trajectories with video frames, while Unified World Models (Zhu et al., 2025a) jointly models video and action diffusion for dynamics and policy learning. Complementary approaches target physical fidelity through physics-aware conditioning (Wang et al., 2025a), simulation-based post-training (Li et al., 2025a), relational alignment (Zhang et al., 2025), latent motion priors (Jiang et al., 2026), and staged motion control (Xu et al., 2026). These approaches primarily address video-level generation or control; we focus on explicitly predicting the outcomes of individual interactions.

Keyframe- and Event-Guided Generation. KeyWorld (Li et al., 2025b) generates task-relevant keyframes from an initial observation and a global task description before reconstructing dense video. SKIP (He et al., 2026) learns event-preserving sparse rollouts with interpolation. DCARL (Ouyang et al., 2026) trains a keyframe generator to establish consistent structural anchors for long trajectory generation. RoboEnvision (Yang et al., 2025a) decomposes manipulation

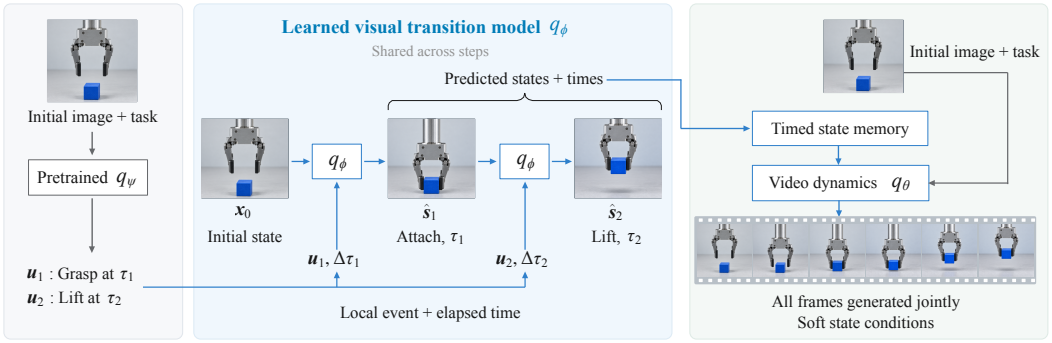


Figure 2: **The overall framework of ~~STRIKE~~.** (a) A pretrained vision-language event planner receives the initial image x_0 and task c, and proposes local events u_k and outcome times τ_k. (b) A shared visual transition model q_ϕ is applied successively using the current state, event, and elapsed time $\Delta\tau_k = \tau_k - \tau_{k-1}$. Observed event-aligned state pairs supervise its training. (c) A separately trained dynamics model q_θ generates the dense rollout from the initial context and soft conditions supplied by the predicted states and their temporal locations.

tasks into atomic instructions, generates aligned boundary keyframes, and fills the intervening segments conditioned on those instructions. CausalMotion (Zhuang et al., 2026) uses VLM-planned keyframes and object trajectories to guide a pretrained video generator without additional training, while CoECT (Wang et al., 2026) constructs event keyframes through event-chain reasoning and iterative editing. We instead directly supervise a shared transition model on event-aligned state pairs, conditioned on the current visual state, one interaction specification, and elapsed time.

Visual Subgoal and State-Transition Prediction. Visual state prediction also supports manipulation representation learning and policy guidance. MPI (Zeng et al., 2024) pretrains representations through interaction-frame prediction and object localization. SuSIE (Black et al., 2024) uses an image-editing diffusion model to propose language-conditioned subgoals for a low-level policy, while TaKSIE (Kang & Kuo, 2025) incorporates task progress for adaptive subgoal generation. Rather than learning representations or subgoals, we supervise interaction-conditioned transitions across physical phenomena and use the timed visual outcomes to condition dense video dynamics.

3 Preliminaries

3.1 Visual States and Transition Conditions

We consider generating a visual rollout $\boldsymbol{X} = (\boldsymbol{x}_0, \boldsymbol{x}_1, \ldots, \boldsymbol{x}_{F-1})$ from an initial visual condition $\boldsymbol{x}_0$ and task context $\boldsymbol{c}$, such as a scene description or a manipulation instruction. For image-conditioned generation, $\boldsymbol{x}_0$ is an observed frame; text-only generation additionally requires synthesizing an initial image before this rollout stage.

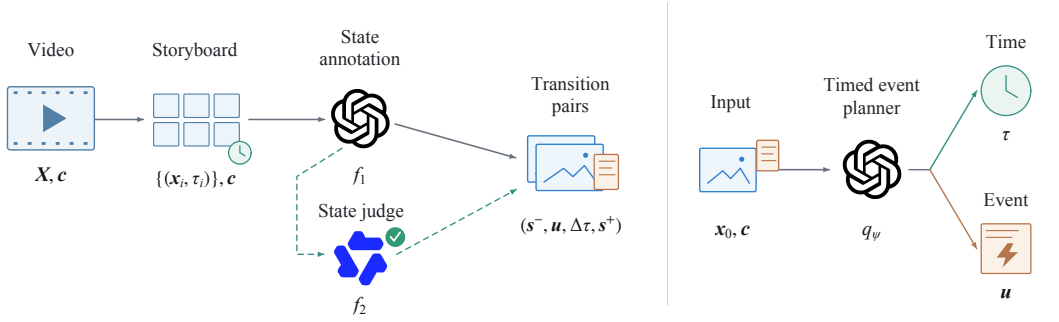
We use a *visual state* to denote an image depicting a scene configuration, including observable object properties and relations such as contact, support, and attachment. A state can depict either an interaction boundary or its persistent consequence, such as first contact with a surface or an object resting after landing. Unlike a complete simulator state, a visual state does not explicitly encode quantities such as velocity or occluded contacts.

We distinguish a visual state s from a *transition specification* u, which describes a local change and its anticipated visible outcome. For example, a release transition may specify that an object becomes detached from a gripper; the corresponding state image depicts how that outcome is realized in the particular scene. The global context c describes the overall task or process, whereas u specifies an individual step.

Let $S = (s_1, \ldots, s_K)$ denote K future visual states at ordered video timestamps $\tau_1, \ldots, \tau_K$. We set $s_0 = x_0$ and $\tau_0 = 0$, and define:

$$0 = \tau_0 < \tau_1 < \cdots < \tau_K, \qquad \Delta\tau_k = \tau_k - \tau_{k-1}. \tag{1}$$

Thus, τ_k locates a state within the rollout, while $\Delta\tau_k$ specifies the interval between successive states. The initial observation is not counted among the K future states.



(a) Training data annotation **(b) Inference: timed event planning**

Figure 3: **Data annotation and timed event planning.** (a) A training video V and caption c are converted into a time-labeled storyboard. GPT-5.5 (OpenAI, 2026) (f_1) selects visual states and annotates their transitions. A Qwen3-VL state judge (f_2) checks image–description agreement, with the dashed branch indicating source-specific filtering. Selected video frames and their annotations form transition pairs $(s^-, u, \Delta\tau, s^+)$. (b) During inference, the event planner q_ψ receives an initial image x_0 and task context c, and jointly predicts transition specifications u and their target times τ, without observing future reference frames.

3.2 Conditional Latent Generation

Conditional image and video generators operate on latent representations of their prediction targets. Let $\boldsymbol{z} = \mathcal{E}(\boldsymbol{X})$ denote the latent encoding of an image or video target $\boldsymbol{X}$, and let $\boldsymbol{C}$ collect the available conditioning information. A general denoising formulation constructs:

$$\boldsymbol{z}_\lambda = a_\lambda \boldsymbol{z} + b_\lambda \boldsymbol{\epsilon}, \qquad \boldsymbol{\epsilon} \sim \mathcal{N}(0, \boldsymbol{I}), \tag{2}$$

and optimizes

$$\mathcal{L}_{\text{gen}}(\eta) = \mathbb{E}_{\boldsymbol{z},\boldsymbol{C},\lambda,\boldsymbol{\epsilon}} \left[w_\lambda \left\| D_\eta(\boldsymbol{z}_\lambda, \lambda; \boldsymbol{C}) - \boldsymbol{z}_\lambda^{\text{tgt}} \right\|_2^2 \right], \tag{3}$$

where a_λ and b_λ define the corruption process, D_η is the conditional predictor, $\boldsymbol{z}_\lambda^{\text{tgt}}$ is its prediction target. w_λ is the backbone-specific noise schedule weighting. The denoising coordinate λ is distinct from both the video timestamp τ_k and the transition interval $\Delta\tau_k$.

Our method retains these generative objectives and introduces structure through the prediction targets and conditioning interface: a transition model predicts sparse visual states under explicit transition and temporal conditions, a video model generates the dense rollout conditioned on those states.

4 Method

As illustrated in Figure 2, our framework contains two learned generative components: a visual state-transition model and a state-conditioned video dynamics model. A pretrained vision-language model (OpenAI, 2026) supplies the semantic and temporal conditions that organize their predictions.

4.1 Data Annotation and Timed Event Planning

We use a common transition interface in two settings: data annotation of training videos and prospective event planning for generation (Figure 3). Annotation describes observed changes to construct supervision; planning predicts the changes to realize from the available initial condition and task context.

Data annotation. For each training video V and its caption c, we sample candidate frames and construct a storyboard labeled with frame indices and timestamps. A first VLM pass identifies the main entities, visible events, and their chronological structure. A second pass selects informative state frames and describes the transitions between adjacent selected states. Selection favors clearly established relations and distinct outcomes, such as contact, deformation, or a settled configuration, rather than generic beginning, middle or end coverage.

We validate candidate identities and temporal order, resolving the selected frames and their timestamps against the candidate table. Quality screening considers event visibility, caption alignment,

state coverage, and frame redundancy. Selected state images are extracted from the training videos, not synthesized by the VLM. Adjacent states produce the transition-pair corpus

$$\mathcal{D}_{\text{trans}} = \{(\boldsymbol{s}^-, \boldsymbol{u}, \Delta\tau, \boldsymbol{s}^+)\}, \tag{4}$$

where u describes the observed change and $\Delta\tau$ is derived from the selected frame indices on the video timeline. Pairs retain a common video-level grouping identifier for train/validation splitting. This corpus provides the event-aligned supervision in Section 4.2. The annotated video/state format also supports the timed-memory supervision in Section 4.3; the two models need not use identical filtered subsets.

Timed event planning. For open-domain inference, a pretrained VLM instead receives only the initial visual condition x_0 and task context c and predicts

$$\hat{\boldsymbol{P}} \sim q_\psi(\boldsymbol{P} \mid \boldsymbol{x}_0, \boldsymbol{c}), \qquad \boldsymbol{P} = \{(\boldsymbol{u}_k, \tau_k)\}_{k=1}^{K}. \tag{5}$$

Each u_k specifies a local change and its anticipated visible outcome, while τ_k locates that outcome within the rollout. Unlike offline annotation, this stage has no access to future reference frames. We convert the plan into the transition model's instruction format, validate temporal order and horizon bounds, and compute $\Delta\tau_k = \tau_k - \tau_{k-1}$. The descriptions and relative intervals condition successive state predictions; the absolute timestamps position those predictions within the video model. The VLM components are not optimized jointly with either generator.

4.2 Learning and Composing Visual State Transitions

We instantiate the transition model q_ϕ with Qwen-Image-Edit (Wu et al., 2025) and fine-tune it on examples $(\boldsymbol{s}^-, \boldsymbol{u}, \Delta\tau, \boldsymbol{s}^+)$. The source image s^- provides the current scene configuration, u specifies the local change, and $\Delta\tau$ supplies the prediction interval. The target s^+ directly supervises the visual configuration resulting from that change.

Relative-time conditioning. We encode elapsed time through the temporal coordinates of the image tokens' rotary positional embeddings (RoPE) (Su et al., 2023), retaining their spatial coordinates. Source and target image tokens receive temporal positions 0 and d, respectively. If a plan specifies frame indices i_{k-1} and i_k on a grid with frame rate f_{plan}, we convert their difference to the canonical training-frame units:

$$d_k = \text{round}\left((i_k - i_{k-1}) \frac{f_{\text{train}}}{f_{\text{plan}}} \right). \tag{6}$$

Here, d_k maintains similar representation as $\Delta\tau_k$, not the denoising coordinate λ. This temporal-position modification introduces no additional trainable parameters.

Transition supervision. Let $\boldsymbol{z}^+ = \mathcal{E}_I(\boldsymbol{s}^+)$ be the target image latent. For Gaussian noise $\boldsymbol{\epsilon}$ and noise level $\lambda \in [0, 1]$, we construct noisy target latent:

$$\boldsymbol{z}_\lambda^+ = (1 - \lambda)\boldsymbol{z}^+ + \lambda\boldsymbol{\epsilon}, \tag{7}$$

and the transition model q_ϕ is trained to optimize the flow-matching objective

$$\mathcal{L}_{\text{trans}}(\phi) = \mathbb{E}\left[\left\| \boldsymbol{v}_\phi(\boldsymbol{z}_\lambda^+, \lambda; \boldsymbol{s}^-, \boldsymbol{s}_0, \boldsymbol{u}, d) - (\boldsymbol{\epsilon} - \boldsymbol{z}^+) \right\|_2^2 \right]. \tag{8}$$

The source image remains uncorrupted, and the loss is applied only to target outputs. We also introduce a first-frame ($\boldsymbol{s}_0$) sink that keeps the latent of the first frame as a persistent visual reference. This helps preserve object identity, scene geometry and appearance over autoregressive transition state generation. We fine-tune the transformer while keeping the pretrained image and multi-modal conditioning encoders frozen.

Sequential composition. We reuse the learned transition model to materialize the plan:

$$\hat{\boldsymbol{s}}_0 = \boldsymbol{x}_0, \qquad \hat{\boldsymbol{s}}_k \sim q_\phi\left(\boldsymbol{s}_k \mid \hat{\boldsymbol{s}}_{k-1}, \hat{\boldsymbol{s}}_0, \boldsymbol{u}_k, \Delta\tau_k\right), \quad k = 1, \ldots, K. \tag{9}$$

The temporal interval is represented internally by d_k from Eqn. (6). Each prediction becomes the visual input to the next transition, with parameters shared across steps. Training uses dataset source images, whereas inference uses the preceding prediction; consequently, state-generation errors can accumulate during composition. Optional terminal hold anchors reuse the preceding image rather than requiring an additional learned transition.

Table 1: **Physics-IQ Verified.** Comparison of model size, generation cost (EFLOPs), and physical consistency across baselines. **Bold** and underlined denote best and second-best scores.

Model	# params	EFLOPs	Score ↑	S-IoU ↑	ST-IoU ↑	WS-IoU ↑
Wan2.2 (Wan Team, 2025)	14B	49.100	34.2	**50.3**	25.2	29.9
Cosmos-Predict-2.5 (Ali et al., 2025)	2B	2.560	32.2	45.5	27.2	30.1
CogVideoX (Yang et al., 2025b)	5B	6.640	30.5	40.4	30.3	24.1
Cosmos3-Nano NVIDIA (2026)	16B	11.802	27.6	39.7	20.2	22.1
CoECT (Wang et al., 2026)	20B+5B	29.600	19.9	27.6	23.2	14.8
Wan2.2-5B (Wan Team, 2025)	5B	2.739	19.4	25.0	19.4	16.3
Open-Sora-v2 (Peng et al., 2025)	11B	5.035	19.0	27.0	21.2	10.1
LTX-Video (HaCohen et al., 2024)	13B	1.152	18.4	31.0	14.1	15.0
CausalMotion (Zhuang et al., 2026)	13B	0.395	12.7	19.0	15.0	8.1
Cosmos3-Super (NVIDIA, 2026)	64B	-	**42.7**	-	-	-
MiniMax-H3 (MiniMax Research, 2026)	33B	-	39.8	-	-	-
Ours (CogVideoX-5B)	20B+5B	10.02	<u>41.9</u>	<u>46.7</u>	**53.5**	**36.0**
Ours (Wan2.2-5B)	20B+5B	4.19	39.7	45.2	<u>53.0</u>	<u>35.4</u>

4.3 Video Dynamics Conditioned on Timed Visual States

The dynamics model generates the complete video from the predicted states and their temporal locations:

$$\hat{\boldsymbol{X}} \sim q_\theta\big(\boldsymbol{X} \mid \boldsymbol{x}_0, \boldsymbol{c}, \{(\hat{\boldsymbol{s}}_k, \tau_k)\}_{k=1}^{K}\big)\,. \tag{10}$$

We instantiate this model with CogVideoX-5B-I2V (Yang et al., 2025b) and Wan2.2-5B-TI2V (Wan Team, 2025), extending their conditioning interfaces to accept multiple state images. Each predicted state is encoded with the video backbone's VAE and projected into spatial memory tokens. The timestamp τ_k is mapped to the backbone's temporal grid, so each memory token carries both spatial and video-time position. Unlike the transition model's relative interval, this position locates the state within the full rollout.

Schematically, video attention accesses the concatenated sequence

$$\boldsymbol{H}_{\text{attn}} = [\boldsymbol{M}_1; \ldots; \boldsymbol{M}_K; \boldsymbol{H}_{\text{video}}], \tag{11}$$

where $\boldsymbol{M}_k$ contains the tokens for state $\hat{\boldsymbol{s}}_k$ and $\boldsymbol{H}_{\text{video}}$ contains the noisy video tokens. Backbone-specific implementations preserve the original text and initial-image conditioning interfaces. CogVideoX processes a joint memory/video token stream, while Wan inserts fixed state memories into each block's self-attention and retains separate text cross-attention. The denoising prediction is read out only from the video-token outputs.

We retain the backbones' native prediction objectives: scheduler-based velocity prediction for CogVideoX and flow matching for Wan. State images act as soft conditions rather than hard constraints on the corresponding output frames. The model generates the full video window jointly, allowing motion between two states to depend on the wider sequence of predicted outcomes, rather than synthesizing each interval independently.

5 Experiments

5.1 Experiment Setup

We evaluate our proposed system on Physics-IQ Verified (Rädsch et al., 2026; Motamed et al., 2026), PhyGenBench (Meng et al., 2025), PISA Experiments (Li et al., 2025a), and our held-out RoboTwin 2.0 (Chen et al., 2025) split discussed in Section A.1. Physics-IQ Verified evaluates whether generated videos exhibit realistic physical dynamics across phenomena such as gravity, collisions, fluids, materials, lighting, and magnetism. PhyGenBench tests physical commonsense across 27 laws spanning mechanics, optics, thermal phenomena, and material properties, assessing key events, temporal order, and overall naturalness. PISA Experiments focuses on object-dropping scenarios in real and simulated environments, measuring whether generated object trajectories and interactions align with expected physical motion. To evaluate the results of Robotwin2.0 validation against baselines, we apply the EWMScore, a 15-metric adaptation of WorldArena (Shang et al., 2026), with more details provided in Section A.5.

Table 2: **PhyGenBench Results.** Comparison of physical-consistency metrics across baselines. **Bold** and underlined denote best and second-best scores.

Method	# param	Mechanics ↑	Optics ↑	Thermal ↑	Material ↑	Average ↑
VideoCrafter2 (Chen et al., 2024)	1.4B	40.00	58.00	28.89	34.17	42.08
LaVie (Wang et al., 2025b)	0.91B	29.17	50.67	23.33	32.50	35.63
Open-Sora v2 (Peng et al., 2025)	11B	55.00	66.67	47.78	50.83	56.25
LTX-Video (HaCohen et al., 2024)	13B	45.83	65.33	43.33	37.50	49.37
DreamWorld (Tan et al., 2026)	1.3B	54.17	64.67	51.11	43.33	54.17
Cosmos3-Nano (NVIDIA, 2026)	16B	57.50	75.33	48.89	46.67	58.75
Cosmos-Predict2.5 (Ali et al., 2025)	2B	60.00	68.00	52.22	43.33	56.87
PhysVid (Pathak et al., 2026)	1.7B	44.17	60.00	38.89	33.33	45.42
CausalMotion Zhuang et al. (2026)	13B	**73.33**	75.33	65.56	64.17	70.21
Wan2.2 (Wan Team, 2025)	5B	50.83	65.33	41.11	40.00	50.83
CogVideoX (Yang et al., 2025b)	5B	50.00	67.33	44.44	51.67	54.79
VideoREPA (Zhang et al., 2025)	5B	44.17	67.33	43.33	44.17	51.25
LaMo (Jiang et al., 2026)	5B	45.83	65.33	46.67	44.17	51.67
Ours (CogVideoX-5B)	20B+5B	69.17	**80.67**	81.11	**76.67**	**76.91**
Ours (Wan2.2-5B)	20B+5B	67.50	79.33	**82.22**	74.17	75.62

5.2 Main Results

Physics-IQ Verified. As shown in Table 1, our method substantially improves physical consistency across both video backbones. It achieves scores of 41.9 with CogVideoX-5B and 39.7 with Wan2.2-5B, surpassing their respective baselines by 11.4 and 20.3 points. Both variants outperform all baselines with reported component metrics on ST-IoU and WS-IoU, with our CogVideoX-based variant reaching 53.5 and 36.0, respectively. Although Wan2.2-14B retains the highest S-IoU, our variants achieve higher overall scores, highlighting improvements beyond spatial agreement alone. Our best variant also surpasses MiniMax-H3 and is only 0.8 points behind Cosmos3-Super. These results demonstrate that augmenting a 5B video backbone with our 20B auxiliary model provides a competitive alternative to increasing the video generator's scale.

PhyGenBench. Table 2 shows that our method improves physical commonsense across all four domains for both video backbones. Our CogVideoX- and Wan-based variants achieve average scores of 76.91 and 75.62, respectively, improving over their corresponding baselines by 22.12 and 24.79 points. They also surpass the strongest baseline CausalMotion by 6.70 and 5.41 points. Our CogVideoX-based variant achieves the highest scores in optics and material properties, while the Wan-based variant leads in thermal phenomena. Although CausalMotion retains the highest mechanics score, these results demonstrate improvements across physical domains and effectiveness across different video backbones.

Table 3: **Pisa-Experiments results**. Results are aggregated over all 421 evaluation videos.

Method	L2 ↓	CD ↓	IoU ↑
CausalMotion	0.1292	0.3435	0.0966
CogVideoX-5B	0.1180	0.3017	0.1481
Cosmos-Predict2.5	0.1373	0.3871	0.1467
Cosmos3-Nano	0.1250	0.3437	0.1361
LaMo-5B	0.1113	0.2888	0.1590
LTX-Video	0.1181	0.3071	0.1017
OpenSoraV2	0.1293	0.3310	0.0846
Ours (CogVideoX-5B)	**0.0949**	**0.2495**	**0.1753**
Ours (Wan2.2-5B)	0.1010	0.2637	0.1485

Pisa-Experiments. As shown in Table 3, our CogVideoX-based variant achieves the lowest L2 and Chamfer Distance, reducing these errors by 19.6% and 17.3%, respectively, relative to CogVideoX-5B-I2V (Yang et al., 2025b). Our Wan-based variant ranks second on both distance metrics, and both variants outperform all evaluated baselines on L2 and CD. Our CogVideoX-based variant also achieves the highest IoU, improving over its CogVideoX-5B backbone. These results indicate that our approach improves geometric accuracy while maintaining competitive spatial overlap across the 421 evaluation videos.

Robotic Manipulation. On RoboTwin2 val500, comprising 500 held-out episodes across 50 tasks and five embodiments under an episode/seed-grouped split, both variants outperform all baselines on every displayed metric (Table 4 & Figure 5). Relative to their respective backbone baselines, our CogVideoX and Wan variants improve EWMScore-15 by 2.47 and 3.37 points, with trajectory-score gains of 43.7% and 42.0%. The Wan variant additionally improves interaction quality and instruction following by 18.2% and 31.8%, reaching 62.53 EWMScore-15. These consistent cross-backbone

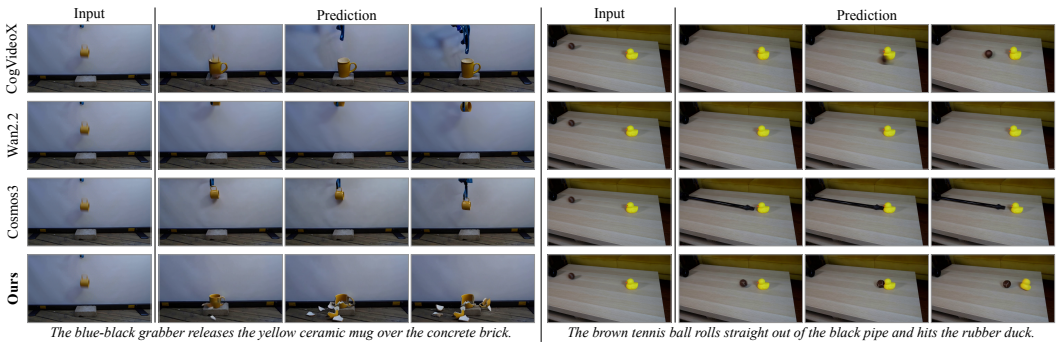


Figure 4: **Qualitative comparisons on Physics-IQ Verified.** Visualizations of four time-aligned frames baselines and ~~STRIKE~~, which depicts mug fracture after impact with a concrete brick (left) and duck displacement following contact with a rolling brown ball (right).

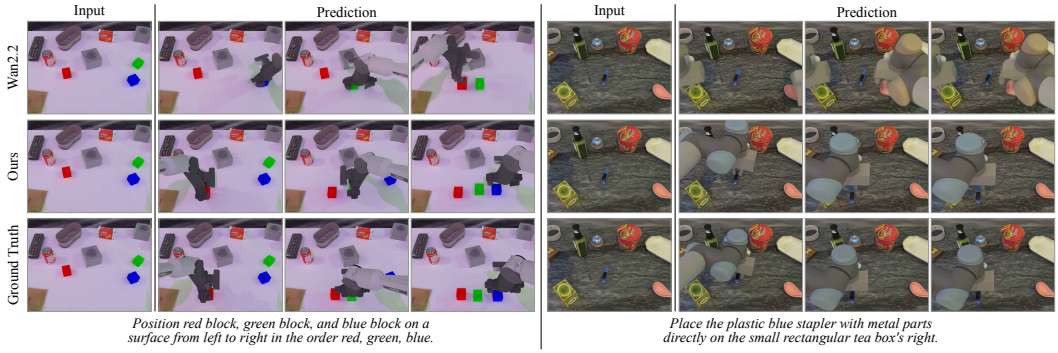


Figure 5: **Qualitative comparisons on RoboTwin2.0.** Visualizations of Wan2.2 (Wan Team, 2025) and ~~STRIKE~~ in comparison against the ground truth.

gains support explicit key-state conditioning for improving trajectory fidelity and task-aligned object interactions in generated manipulation videos.

Qualitative Results. Figure 4 compares our method with the baselines on two Physics-IQ Verified examples. In the mug-drop example, our method captures impact followed by fragmentation, whereas the baselines retain an intact mug or leave it suspended above the concrete block. In the ball–duck example, our sequence depicts the ball approaching the duck, making contact, and inducing displacement. The baselines instead exhibit inconsistent ball motion, missing interactions, or spurious changes to the pipe. By providing explicit intermediate states at planned event times, our method guides video generation toward the intended physical progression.

5.3 Ablation Studies

Learned Transition Model. We compare pretrained Qwen-Image-Edit (QIE) with our trained transition model through the event-planning and transition-state generation pipeline (Figure 6; Table 6). Metrics are computed against reference states, excluding the initial input image. On validation states, PSNR increases from 17.90 to 24.88 and SSIM from 0.674 to 0.863, while LPIPS and L1 decrease by 57.6% and 63.5%, respectively. Both perceptual similarity and pixel-level accuracy improve. These gains accompany improvements in downstream video generation: Physics-IQ rises from 32.60 to 41.93, PhyGenBench from 72.50 to 76.91, and Pisa-IoU from 0.1224 to 0.1753. Together, these results support learning task-specific transitions and using their predicted states to condition dense video synthesis.

Design Choice Analysis. Table 5 examines first-frame anchoring, temporal conditioning, and synthetic data. Removing the first-frame anchor lowers Physics-IQ/Pisa-IoU to 40.60/0.1528, supporting persistent conditioning on the initial visual reference during recursive prediction. Without time conditioning, Physics-IQ decreases modestly to 41.63, whereas Pisa-IoU falls to 0.1617, indicating benchmark-dependent benefits. Excluding synthetic data yields 37.91/0.1523 and the largest

Table 4: **Comparison on the RoboTwin2 val500 benchmark.** EWMScore-15 is the arithmetic mean of all 15 metrics. The best result is in **bold**, and the second-best result is underlined.

Method	EWMScore ↑	Trajectory ↑	Interaction ↑	Perspectivity ↑	Instruction ↑	Semantic ↑
OpenDW	44.26	0.0220	0.2556	0.6096	0.2080	0.8685
Ctrl-World	60.16	0.2217	0.5268	0.7776	0.4960	0.8782
CogVideoX-5B	58.04	0.2301	0.5432	0.7952	0.5180	0.8934
Wan2.2-5B	59.16	0.2372	0.5088	0.8040	0.4560	0.8866
Ours (CogVideoX-5B)	60.51	0.3306	0.5900	0.8160	0.5916	**0.8975**
Ours (Wan2.2-5B)	**62.53**	**0.3368**	**0.6012**	**0.8448**	**0.6012**	0.8943

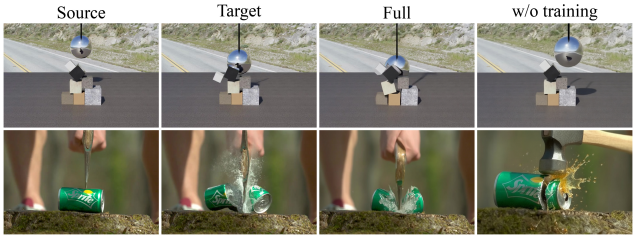


Figure 6: Qualitative comparison between our trained transition model and the pretrained Qwen Image Edit.

Table 5: Ablation on transition model design choices.

Variants	Physics-IQ↑	Pisa-IoU↑
w/o sink	40.60	0.1528
w/o time	41.63	0.1617
w/o synthetic	37.91	0.1523
Full model	**41.93**	**0.1753**

Table 6: Ablation Studies on learning transition model.

	State Transition Evaluation				Video Generation Evaluation		
Model	PSNR↑	SSIM↑	LPIPS↓	L1↓	Physics-IQ↑	PhyGenBench↑	Pisa-IoU↑
w/o training	17.90	0.674	0.436	0.104	32.60	72.50	0.1224
Ours	**24.88**	**0.863**	**0.185**	**0.038**	**41.93**	**76.91**	**0.1753**

Physics-IQ drop of 4.02 points. These comparisons favor retaining all three design choices, although differences in training mixtures and budgets limit strict causal attribution to individual components.

6 Conclusion and Limitation

We presented a physical world modeling framework that explicitly learns visual state transitions and uses their timed predictions to condition dense video generation. VLM-assisted annotation constructs event-aligned supervision, while a pretrained event planner provides transition specifications for open-domain inference. Across physical-video and robotic manipulation benchmarks, the resulting intermediate-state representation improves physical-consistency and manipulation- video metrics over corresponding video backbones. However, the framework remains a learned visual predictor rather than an explicit physics simulator: plausible states do not guarantee correct trajectories, contact dynamics, or conservation laws. Since training uses observed states but inference conditions on recursively predicted states, errors may accumulate over long horizons. The pipeline also introduces additional planning and image-generation costs and depends on the quality of model-assisted annotations.

AI use statement

In our method, we used GPT-5.5 for training-data event annotation and inference-time event planning, Qwen3-VL for selected state-consistency checks, Qwen-Image-Edit for visual state-transition generation, and CogVideoX/Wan for conditional video synthesis. We also used AI for manuscript and supplementary material polishing and code assistance, including experiments execution and establishment of supplementary presentation. We have not used AI to fabricate, alter, or selectively report experimental measurements; generate benchmark ground-truth data; create undisclosed citations; or make final scientific decisions without author review. All AI-assisted material was reviewed by the authors. We take full responsibility for the final content of this work.

Reproducibility statement

We describe the problem formulation in Sections 3 and 4. The data sources, curation procedure, dataset composition, are documented in Appendix A.1. The training configurations, hyperparameters and training compute are listed with details in Section A.3 and Section A.4

References

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, Zongyu Lin, Tianyi Xie, Zeshun Zong, Michal Yarom, Yonatan Bitton, Chenfanfu Jiang, Yizhou Sun, Kai-Wei Chang, and Aditya Grover. Videophy: Evaluating physical commonsense for video generation. In *International Conference on Learning Representations*, volume 2025, pp. 102075–102121, 2025.

Kevin Black, Mitsuhiko Nakamoto, Pranav Atreya, Homer Walke, Chelsea Finn, Aviral Kumar, and Sergey Levine. Zero-shot robotic manipulation with pre-trained image-editing diffusion models. In *The Twelfth International Conference on Learning Representations*, 2024.

Haoxin Chen, Yong Zhang, Xiaodong Cun, Menghan Xia, Xintao Wang, Chao Weng, and Ying Shan. Videocrafter2: Overcoming data limitations for high-quality video diffusion models. In *2024 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 7310–7320. IEEE, 2024.

Tianxing Chen, Zanxin Chen, Baijun Chen, Zijian Cai, Yibin Liu, Zixuan Li, Qiwei Liang, Xianliang Lin, Yiheng Ge, Zhenyu Gu, et al. Robotwin 2.0: A scalable data generator and benchmark with strong domain randomization for robust bimanual robotic manipulation. *arXiv preprint arXiv:2506.18088*, 2025.

Yilun Du, Sherry Yang, Bo Dai, Hanjun Dai, Ofir Nachum, Josh Tenenbaum, Dale Schuurmans, and Pieter Abbeel. Learning universal policies via text-guided video generation. In A. Oh, T. Naumann, A. Globerson, K. Saenko, M. Hardt, and S. Levine (eds.), *Advances in Neural Information Processing Systems*, volume 36, pp. 9156–9172. Curran Associates, Inc., 2023. doi: 10.52202/075280-0403. URL https://proceedings.neurips.cc/paper_files/paper/2023/file/1d5b9233ad716a43be5c0d3023cb82d0-Paper-Conference.pdf.

Frederik Ebert, Chelsea Finn, Sudeep Dasari, Annie Xie, Alex Lee, and Sergey Levine. Visual foresight: Model-based deep reinforcement learning for vision-based robotic control, 2018. URL https://arxiv.org/abs/1812.00568.

Yoav HaCohen, Nisan Chiprut, Benny Brazowski, Daniel Shalem, Dudu Moshe, Eitan Richardson, Eran Levin, Guy Shiran, Nir Zabari, Ori Gordon, Poriya Panet, Sapir Weissbuch, Victor Kulikov, Yaki Bitterman, Zeev Melumian, and Ofir Bibi. Ltx-video: Realtime video latent diffusion. *arXiv preprint arXiv:2501.00103*, 2024.

Danijar Hafner, Timothy Lillicrap, Ian Fischer, Ruben Villegas, David Ha, Honglak Lee, and James Davidson. Learning latent dynamics for planning from pixels. In *Proceedings of the 36th International Conference on Machine Learning*, volume 97, pp. 2555–2565. PMLR, 2019. URL https://proceedings.mlr.press/v97/hafner19a.html.

Danijar Hafner, Timothy Lillicrap, Jimmy Ba, and Mohammad Norouzi. Dream to control: Learning behaviors by latent imagination. In *International Conference on Learning Representations*, 2020. URL https://openreview.net/forum?id=S1lOTC4tDS.

Ziheng He, Yixiang Chen, Ning Yang, Zhanqian Wu, Qisen Ma, Yuan Xu, Jiabing Yang, Peiyan Li, Xiangnan Wu, Xiaofeng Wang, Zheng Zhu, Jing Liu, Nianfeng Liu, and Yan Huang. SKIP: Sparse keyframe interpolation paradigm for efficient embodied world models, 2026. URL https://arxiv.org/abs/2606.00664v1.

Bo Jiang, Depu Meng, Yihan Hu, Yichen Xie, Tianshuo Xu, and Wei Zhan. Lamo: Self-supervised latent motion priors for physical realism in video generation. *arXiv preprint arXiv:2605.23878*, 2026.

Xuhui Kang and Yen-Ling Kuo. Incorporating task progress knowledge for subgoal generation in robotic manipulation through image edits. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, pp. 7490–7499, 2025. doi: 10.1109/WACV61041.2025.00728.

Chenyu Li, Oscar Michel, Xichen Pan, Sainan Liu, Mike Roberts, and Saining Xie. Pisa experiments: Exploring physics post-training for video diffusion models by watching stuff drop. In *Proceedings of the 42nd International Conference on Machine Learning*, volume 267 of *Proceedings of Machine Learning Research*, pp. 35685–35709. PMLR, 2025a. URL https://proceedings.mlr.press/v267/li25bu.html.

Sibo Li, Qianyue Hao, Yu Shang, and Yong Li. KeyWorld: Key frame reasoning enables effective and efficient world models, 2025b. URL https://arxiv.org/abs/2509.21027.

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

Fanqing Meng, Jiaqi Liao, Xinyu Tan, Quanfeng Lu, Wenqi Shao, Kaipeng Zhang, Yu Cheng, Dianqi Li, and Ping Luo. Towards world simulator: Crafting physical commonsense-based benchmark for video generation. In *Proceedings of the 42nd International Conference on Machine Learning*, volume 267 of *Proceedings of Machine Learning Research*, pp. 43781–43806. PMLR, 2025. URL https://proceedings.mlr.press/v267/meng25c.html.

MiniMax Research. Minimax h3: An open model breaking the boundaries between tasks and modalities. https://www.minimax.io/blog/minimax-h3, July 2026.

Saman Motamed, Laura Culp, Kevin Swersky, Priyank Jaini, and Robert Geirhos. Do generative video models understand physical principles? In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, pp. 948–958, 2026. doi: 10.1109/WACV61042.2026.00099.

Kepan Nan, Rui Xie, Penghao Zhou, Tiehan Fan, Zhenheng Yang, Zhijie Chen, Xiang Li, Jian Yang, and Ying Tai. OpenVid-1M: A large-scale high-quality dataset for text-to-video generation. In *International Conference on Learning Representations*, 2025.

Sriram Narayanan, Ziyu Jiang, Srinivasa G. Narasimhan, and Manmohan Chandraker. Phyco: Learning controllable physical priors for generative motion. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 41892–41902, 2026.

NVIDIA. Cosmos 3: Omnimodal world models for physical ai. *arXiv preprint arXiv:2606.02800*, 2026. URL https://arxiv.org/abs/2606.02800.

NVIDIA. Physicalai-worldmodel-synthetic-physical-interaction-scenes, 2026. URL https://huggingface.co/datasets/nvidia/PhysicalAI-WorldModel-Synthetic-Physical-Interaction-Scenes.

OpenAI. GPT-5.5. https://openai.com/index/introducing-gpt-5-5/, April 2026. Accessed: 2026-09-19.

Junyi Ouyang, Wenbin Teng, Gonglin Chen, Yajie Zhao, and Haiwei Chen. Dcarl: A divide-and-conquer framework for autoregressive long-trajectory video generation. In *European Conference on Computer Vision*, pp. 610–626. Springer, 2026.

Saurabh Pathak, Elahe Arani, Mykola Pechenizkiy, and Bahram Zonooz. Physvid: Physics aware local conditioning for generative video models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 41847–41858, 2026.

Xiangyu Peng, Zangwei Zheng, Chenhui Shen, Tom Young, Xinying Guo, Binluo Wang, Hang Xu, Hongxin Liu, Mingyan Jiang, Wenjun Li, Yuhui Wang, Anbang Ye, Gang Ren, Qianran Ma, Wanying Liang, Xiang Lian, Xiwen Wu, Yuting Zhong, Zhuangyan Li, Chaoyu Gong, Guojun Lei, Leijun Cheng, Limin Zhang, Minghao Li, Ruijie Zhang, Silan Hu, Shijie Huang, Xiaokang Wang, Yuanheng Zhao, Yuqi Wang, Ziang Wei, and Yang You. Open-sora 2.0: Training a commercial-level video generation model in $200k. *arXiv preprint arXiv:2503.09642*, 2025.

Yuandong Pu, Le Zhuo, Songhao Han, Jinbo Xing, Kaiwen Zhu, Shuo Cao, Bin Fu, Si Liu, Hongsheng Li, Yu Qiao, Wenlong Zhang, Xi Chen, and Yihao Liu. Picabench: How far are we from physical realistic image editing? In *The Fourteenth International Conference on Learning Representations*, 2026.

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.

Yu Shang, Zhuohang Li, Yiding Ma, Weikang Su, Xin Jin, Ziyou Wang, Lei Jin, Xin Zhang, Yinzhou Tang, Haisheng Su, et al. Worldarena: A unified benchmark for evaluating perception and functional utility of embodied world models. *arXiv preprint arXiv:2602.08971*, 2026.

Jianlin Su, Yu Lu, Shengfeng Pan, Ahmed Murtadha, Bo Wen, and Yunfeng Liu. Roformer: Enhanced transformer with rotary position embedding, 2023. URL https://arxiv.org/abs/2104.09864.

Boming Tan, Xiangdong Zhang, Ning Liao, Yuqing Zhang, Shaofeng Zhang, Xue Yang, Qi Fan, and Yanyong Zhang. Dreamworld: Unified world modeling in video generation. *arXiv preprint arXiv:2603.00466*, 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.

Wan Team. Wan2.2. Official model repository, 2025. URL https://github.com/Wan-Video/Wan2.2.

Jing Wang, Ao Ma, Ke Cao, Jun Zheng, Jiasong Feng, Zhanjie Zhang, Wanyuan Pang, and Xiaodan Liang. Wisa: World simulator assistant for physics-aware text-to-video generation. In *Advances in Neural Information Processing Systems*, volume 38, pp. 5388–5416, 2025a.

Yaohui Wang, Xinyuan Chen, Xin Ma, Shangchen Zhou, Ziqi Huang, Yi Wang, Ceyuan Yang, Yinan He, Jiashuo Yu, Peiqing Yang, et al. Lavie: High-quality video generation with cascaded latent diffusion models. *International Journal of Computer Vision*, 133(5):3059–3078, 2025b.

Zixuan Wang, Yixin Hu, Haolan Wang, Feng Chen, Yan Liu, Wen Li, and Yinjie Lei. Chain of event-centric causal thought for physically plausible video generation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 38122–38131, 2026.

Chenfei Wu, Jiahao Li, Jingren Zhou, Junyang Lin, Kaiyuan Gao, Kun Yan, Sheng-ming Yin, Shuai Bai, Xiao Xu, Yilei Chen, et al. Qwen-image technical report. *arXiv preprint arXiv:2508.02324*, 2025.

Tianshuo Xu, Zhifei Chen, Leyi Wu, Hao Lu, and Ying-cong Chen. Motion forcing: A decoupled framework for robust video generation in motion dynamics, 2026. URL https://arxiv.org/abs/2603.10408.

Liudi Yang, Yang Bai, George Eskandar, Fengyi Shen, Mohammad Altillawi, Dong Chen, Soumajit Majumder, Ziyuan Liu, Gitta Kutyniok, and Abhinav Valada. RoboEnvision: A long-horizon video generation model for multi-task robot manipulation. In *2025 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pp. 21281–21288, 2025a. doi: 10.1109/IROS60139.2025.11246352.

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

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*, 2025b.

Jia Zeng, Qingwen Bu, Bangjun Wang, Wenke Xia, Li Chen, Hao Dong, Haoming Song, Dong Wang, Di Hu, Ping Luo, Heming Cui, Bin Zhao, Xuelong Li, Yu Qiao, and Hongyang Li. Learning manipulation by predicting interaction. In *Proceedings of Robotics: Science and Systems*, 2024. doi: 10.15607/RSS.2024.XX.123.

Xiangdong Zhang, Jiaqi Liao, Shaofeng Zhang, Fanqing Meng, Xiangpeng Wan, Junchi Yan, and Yu Cheng. Videorepa: Learning physics for video generation through relational alignment with foundation models. In *Advances in Neural Information Processing Systems*, volume 38, pp. 122647–122676, 2025.

Chuning Zhu, Raymond Yu, Siyuan Feng, Benjamin Burchfiel, Paarth Shah, and Abhishek Gupta. Unified world models: Coupling video and action diffusion for pretraining on large robotic datasets. In *Proceedings of Robotics: Science and Systems*, 2025a. doi: 10.15607/RSS.2025.XXI.015.

Fangqi Zhu, Hongtao Wu, Song Guo, Yuxiao Liu, Chilam Cheang, and Tao Kong. IRASim: A fine-grained world model for robot manipulation. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 9834–9844, 2025b.

Sihan Zhuang, Xinyuan Chen, Tianfan Xue, and Yaohui Wang. CausalMotion: Structured physical reasoning as keyframe and trajectory guidance for training-free video generation, 2026. URL https://arxiv.org/abs/2606.14317v1.

A Additional Training Details

A.1 Datasets

We use complementary datasets for learning visual state transitions, training conditional video dynamics, and studying robotic manipulation. Table 7 summarizes the prepared corpora. The reported sizes refer to curated data before reserving internal validation samples, rather than to the nominal sizes of the upstream datasets.

WISA. WISA (Wang et al., 2025a) provides real-world videos covering diverse physical phenomena. Our transition corpus retains 59,298 episodes across 17 annotated categories, including collisions, elastic motion, deformation, liquid and gas motion, phase changes, combustion, reflection, and refraction. WISA contributes 160,672 adjacent-state pairs and is also a source for the separately curated video-dynamics corpus described below.

NVIDIA PhysicalAI. We use 39,873 episodes from NVIDIA PhysicalAI's synthetic physical-interaction scenes (NVIDIA, 2026). Our selected subset covers six scenario families: billiards, bowling, dominoes, falling objects, objects rolling on ramps, and wrecking-ball interactions. These simulated videos provide examples of contact, momentum transfer, falling, and chained rigid-body interactions that complement the visual diversity of WISA. Annotation produces 116,358 adjacent-state pairs from this source; we use the selected subset, not the entire upstream PhysicalAI collection.

PhyCo/Kubric. We additionally curate 31,109 episodes from PhyCo's Kubric-based simulations (Narayanan et al., 2026). The prepared subset comprises four scenario families: rigid ball drops, soft-body ball drops, ball–wall collisions, and deformable-cube interactions. It contributes 76,810 adjacent-state pairs, providing supervision for changes around impact, rebound, and deformation. Together, WISA, PhysicalAI, and PhyCo/Kubric supply 130,280 video episodes and 353,840 event-aligned transition pairs.

PICA-100K. PICA-100K (Pu et al., 2026) supplies auxiliary instruction-based image-editing supervision. It consists of synthetic, video-derived editing examples rather than trajectories from a physics simulator. Our prepared manifest contains 105,085 source–target image pairs, each paired with an editing instruction. These examples extend supervision to instruction-driven changes without requiring a temporal annotation. We therefore use them as text-conditioned image pairs with the time condition omitted, rather than assigning an artificial elapsed time or counting them as additional video episodes.

OpenVid-1M and conditional video dynamics. OpenVid-1M (Nan et al., 2025) provides captioned, open-domain videos that complement WISA for training the conditional dynamics model. We annotate sparse states and retain WISA/OpenVid sequences for which all selected frames pass the state-description consistency check described in Section 4.1. This additional filtering is specific to this video-training path and is not shared by all transition-pair sources. The prepared CogVideoX (Yang et al., 2025b) corpus contains 69,344 episodes (35,114 WISA and 34,230 OpenVid), represented as 49-frame clips at 720×480. The Wan2.2 (Wan Team, 2025) corpus contains same number of episodes, represented as 81-frame clips at 832×480. Each backbone reserves 2,000 episodes for internal validation. Training uses observed state images with their temporal locations as conditions and the corresponding full video as the target.

RoboTwin2.0. For robotic manipulation, we use 124,650 complete simulated episodes from RoboTwin2.0 (Chen et al., 2025), spanning 50 tasks and five robot embodiments: ALOHA-AgileX, ARX-X5, Franka, UR5, and Piper. This snapshot contains 11,150 clean and 113,500 domain-randomized episodes. We construct a deterministic 90/10 split into 112,185 training and 12,465 validation episodes, keeping episodes with the same task and simulation seed in the same split across embodiments and clean/randomized configurations. This prevents the same task–seed group from appearing on both sides. For the reported *val500* evaluation, we select 500 held-out episodes covering all 50 tasks and all five embodiments; every selected episode is domain-randomized. Thus, the evaluation measures generalization to unseen episodes of known tasks, not to held-out task categories.

Table 7: Prepared datasets used in our experiments.

Dataset	Type	Size
Transition-model supervision		
WISA-80K	Real	59,298 episodes
NVIDIA PhysicalAI	Simulation	39,873 episodes
PhyCo/Kubric	Simulation	31,109 episodes
PICA-100K	Synthetic	105,085 image pairs
Conditional video-dynamics supervision		
WISA-80K	Real	35,114 episodes
OpenVid-1M	Real	34,230 episodes
Robotic manipulation		
RoboTwin2.0	Simulation	112,185 train; 12,465 val
RoboTwin2.0 val500	Simulation	500 validation episodes

A.2 Event Planning and Data Annotation Pipeline

We construct transition state frame annotations through a two-stage vision–language model (VLM) annotation pipeline, followed by VQA-based verification. For each video, candidate frames are arranged into a temporally ordered storyboard and provided alongside the video caption, available source metadata, and a candidate table specifying frame identifiers and timestamps.

Stage I: Event Planning. In (Table 8), the annotator identifies the main objects and physical or visual process, characterizes the event structure, and decomposes the observed progression into a minimal sequence of visible subevents. It then constructs a state plan grounded in candidate frames, prioritizing physical state changes while also allowing camera motion that reveals new content or distinct viewpoints. Clips without useful visible state or view progression are flagged as unsuitable for event-based annotation.

Stage II: Transition State Frame Selection and Annotation. In (Table 10), the annotator uses this event plan to select a compact, chronologically ordered set of event-defining transition state frames, rather than generic beginning–middle–end coverage. Selection prioritizes the earliest frame in which a meaningful state or relation is clearly established, while avoiding near-duplicates and redundant intermediate frames. Each selected transition state frame receives a state description, a selection rationale, and a confidence score; adjacent transition state frames additionally receive a textual description of the intervening transition.

Stage III: Transition State Frame State Verification. In (Table 11) uses Qwen3-VL (Bai et al., 2025) as a separate VQA judge to assess whether each annotated state is supported by the visual evidence. The first transition state frame is evaluated on its own, whereas subsequent transition state frames are evaluated together with the immediately preceding transition state frame as a reference for changes or motion. Conditioned on the video caption, physical category, and claimed state, the judge returns a binary match verdict, a confidence score, and a short justification, rejecting claims that are not clearly visible. These verdicts are aggregated at the video level to support downstream annotation-quality filtering.

A.3 Training Configurations

For transition model, we fine-tune Qwen-Image-Edit (Wu et al., 2025) as a time-conditioned Markov model using paired examples from the comprehensive multi-source training corpus. Each example consists of the previous ground-truth state image, the next state image, and a text instruction describing their transition. Images are encoded with a frozen VAE, while the frozen vision-language encoder processes the instruction and conditioning image. The transformer receives concatenated noisy target latents and clean source latents. Relative transition state frame timing is represented through temporal rotary positional embeddings, assigning positions zero and (Δt) to source and target tokens, respectively. We optimize a flow-matching MSE objective on target tokens only, keeping

Table 8: **Event planning prompt.** Prompt used to identify the visible physical process, ordered subevents, and event-relevant states before transition state selection.

```
You are labeling transition state frames for training a physics-aware
video generation model.

You will receive {N} storyboard images, one per sample.  Each
storyboard image contains labeled candidate frames.

For each sample, do this in order:

 1. Determine whether the video contains a visible physical state
    change.
 2. Identify the main objects, scene elements, and main physical or
    visual process.
 3. Infer event_structure using a concise general label, such as
    single_transition, sequential_transitions, repeated_motion,
    camera_motion, scene_change, or unclear.
 4. Write ordered_subevents as the minimal ordered list of visible
    subevents.  Merge simultaneous changes into one subevent and
    split clearly sequential changes into separate subevents.
 5. Build state_plan around moments that matter for transition state
    selection.  State descriptions should mention visible states or
    relations such as onset of motion, contact or collision,
    deformation, liquid or granular flow, opening or closing,
    placement, stacking, release, stable result, camera reveal, or
    view change.
 6. Reject clips that are mostly static and have no useful visible
    state/view progression.

Return strict JSON only with one item per sample_id.
```

the source image uncorrupted. The complete transformer is fine-tuned with AdamW (Loshchilov & Hutter, 2019) at a learning rate of 10^{-5}, using 100 warm-up steps. During auto-regressive inference, each generated transition state images becomes the semantic condition for the next transition. For conditional dynamic model training, we fine-tune CogVideoX-5B-I2V (Yang et al., 2025b) and Wan2.2-TI2V-5B (Wan Team, 2025) using AdamW (Loshchilov & Hutter, 2019) with learning rate equals to 10^{-5} and weight decay 0.01, keeping the VAE and text encoder frozen. Transition states latents are patchified using the pretrained projection, and temporally aligned with video tokens. The tokens are prepended with video stream for joint text–frame–video attention. Only video outputs are supervised, using velocity prediction for CogVideoX and flow matching for Wan. The transition model is trained with 64 A100 GPUs for 5 epochs with per-GPU batch size of 1 and the conditional dynamics model is trained with 32 GPUs for 5000 steps with per-GPU batch size of 2.

A.4 Training and Inference Protocol

The transition and dynamics models are trained separately. For dynamics training, conditioning states are extracted from the training video and paired with their temporal indices; the dense video supplies the prediction target. At inference, these observed state conditions are replaced by the transition model's generated states. Thus, the training and inference interfaces agree in their structure, while the quality of the state conditions can differ.

Inference consists of state event planning, successive state generation, and dense video sampling conditioned on the resulting timed memories. The original task context is retained by the video generator. Generated videos are subsequently formatted for benchmark evaluation. The benchmark evaluator is external to the generative model and supplies no gradients to either learned component.

Table 9: **Results on VideoPhy.** Model size, full-benchmark generation cost (EFLOPs), and semantic adherence and physical common sense metrics are compared across multiple baseline methods, with **bold** and underlined values indicating the best and second-best scores, respectively. Our results are reported with text-to-video (T2V) setting.

			Solid–Solid		Solid–Fluid		Fluid–Fluid		Overall	
Method	# params	EFLOPs↓	SA ↑	PC ↑	SA ↑	PC ↑	SA ↑	PC ↑	SA ↑	PC ↑
VideoCrafter2	1.4B	0.441	27.3	25.2	47.3	24.7	49.1	30.9	39.2	25.9
Open-Sora v2	11B	9.471	53.9	13.3	72.6	24.0	65.5	27.3	63.7	20.2
LTX-Video	13B	1.379	39.9	17.5	70.6	22.6	49.1	25.5	54.4	20.9
DreamWorld	1.3B	9.830	58.7	12.6	74.0	27.4	70.9	32.7	67.2	22.1
Cosmos3-Nano	16B	14.033	**67.1**	13.3	**82.2**	23.3	78.2	25.5	**75.3**	19.5
Cosmos-Predict2.5	2B	4.448	55.2	20.3	74.7	26.0	74.6	40.0	66.6	25.9
WISA	14B	59.272	58.0	14.7	78.8	18.5	**80.0**	25.5	70.5	18.0
PhysVid	1.7B	11.090	45.5	**21.0**	74.7	30.1	76.4	40.0	62.8	27.9
Wan2.2	5B	1.753	52.5	14.7	76.0	20.6	61.8	25.5	64.0	18.9
CausalMotion	13B	0.502	49.7	9.8	65.8	17.1	63.6	20.0	58.7	14.5
CoECT	20B+5B	-	61.5	9.8	80.1	17.8	69.1	27.3	70.6	16.0
CogVideoX	2B	4.236	51.8	14.7	75.3	30.1	70.9	54.6	64.8	27.6
VideoREPA	2B	4.236	51.8	12.6	76.0	29.5	67.3	56.4	64.5	26.7
LaMo	2B	4.236	58.7	16.8	74.7	**32.2**	69.1	67.3	67.2	**31.4**
Ours (CogVideoX-2B-T2V)	20B+2B	10.170	59.4	19.6	78.1	**32.2**	78.2	**69.1**	70.4	**32.9**

A.5 Robotic Manipulation Evaluation

We evaluate robotic manipulation videos using a 15-metric adaptation of WorldArena (Shang et al., 2026), denoted EWMScore-15. The retained metrics cover visual quality (image quality, aesthetic quality, and JEPA similarity), motion quality (dynamic degree, flow score, and motion smoothness), content consistency (subject, background, and photometric consistency), physics adherence (interaction quality and trajectory accuracy), 3D accuracy (depth accuracy and perspectivity), and controllability (instruction following and semantic alignment). We exclude Action Following, which requires generating videos under multiple alternative instructions from the same initial frame and is outside our single-instruction evaluation protocol. Using the evaluator's metric-specific normalization and aggregation rules, we obtain dataset-level scores $s_j \in [0, 1]$, with higher values indicating better performance, and compute $\mathrm{EWMScore\text{-}15} = \frac{100}{15} \sum_{j=1}^{15} s_j$. All retained metrics receive equal weight. JEPA similarity is evaluated over the complete generated and reference video collections rather than averaged over individual episodes.

B Additional Results on Benchmarks

Table 12 and table 9 demonstrate additional quantitative results on PhyGenBench (Meng et al., 2025) and VideoPhy (Bansal et al., 2025), another benchmark that evaluates physical commonsense. Compared with other baseline methods, our method uses a substantially larger transition model (20B) together with only a lightweight 2B video backbone, yet remains competitive with, and often surpasses, baselines using 11–16B video models. Since the 2B version of CogVideoX (Yang et al., 2024) doesn't have an image-to-video (I2V) branch, our results are reported with text-to-video (T2V) setting. On PhyGenBench (Meng et al., 2025), it achieves the second-best average score (65.68) and the best Material score (66.04). On VideoPhy (Bansal et al., 2025), it obtains the best overall physical common-sense precision (32.9), the best Fluid–Fluid precision (69.1), and competitive overall semantic accuracy (70.4), while using a lightweight 2B video generator. This suggests that allocating capacity to explicit transition-state prediction can be more effective than scaling the video backbone alone: the transition model supplies structured, physically meaningful intermediate states, allowing a compact dynamics model to render temporally coherent videos at substantially lower model complexity. In addition, Figure 7 visualizes qualitative comparisons with our baseline methods.

Table 10: **Transition state frame selection prompt.** Prompt used to select event-relevant transition state frame from candidate frames.

```
Select transition state frames from the candidate frames for each
sample.

Decision procedure:

  • First read the caption, source metadata,
    event_plan.event_structure, event_plan.ordered_subevents, and
    event_plan.state_plan.
  • Choose transition state frames for distinct visible states in the
    main event, not for generic beginning/middle/end coverage.
  • For a single clear transition, choose the earliest candidate where
    the defining changed state is visible; include an earlier setup
    state only when it is needed to understand the change.
  • For sequential transitions, select one frame per visible subevent
    at the earliest candidate where that subevent's state is clearly
    established.
  • For contact, collision, attachment, containment, placement,
    stacking, opening, or closing, choose the first candidate where
    the relation is visibly established, not a later redundant proof
    frame.
  • For deformation, breakage, spilling, pouring, smoke, fire, liquid,
    granular motion, or other continuous processes, cover the onset
    and the most informative result or peak state.  Add an
    intermediate keyframe only when the process has a materially
    different middle state.
  • For object motion, choose frames where the object's state or
    relation has changed meaningfully.  Avoid frames that differ only
    by a small position shift unless that shift completes the event.
  • For camera motion without object state change, choose frames that
    capture distinct viewpoints or newly revealed content.

Rules:

  • Choose only candidate_id values in the sample's candidate table.
  • Do not select static initial frames, near-duplicates, or redundant
    middle frames unless they represent a distinct visible state.
  • A transition state frame should be an event-defining visual state:
    onset, contact, relation established, state transformed, view
    changed, peak action, or stable result.
  • Selected frames must be chronological.
  • Each selected frame must represent a distinct event-relevant
    visible state or relation.
  • For each transition state frame provide:  slot, candidate_id,
    frame_index, time_sec, state_text, selection_reason, confidence.
  • Also provide transition_text between adjacent transition state
    frames.
  • In coverage_notes, explicitly write:
    event_structure=<label>;
    selected_subevents= <comma-separated selected visible states>;
    omitted= <why any expected state was not selected or 'none'>.
```

Table 11: **Transition state frame verification prompt.** Prompt used by Qwen3-VL to judge whether an extracted transition frame state visibly matches its annotated physical state.

System prompt

```
You are a strict quality inspector for a physics-focused video
dataset.  Judge only what is actually visible in the given
frame(s).  Be skeptical:  if the described state is not clearly
visible, answer false.  Always reply with a single JSON object and
nothing else.
```

User prompt for the first transition state frame

```
This image is transition state frame {slot_pos} of {total}
extracted from a video.
Video context:  physical category = ``{label}''; caption =
``{caption}''
The transition state frame annotation claims the scene has reached
this state:  ``{state_text}''
Question:  does the image show that the scene has actually reached
the claimed state?  Judge the physical state of the scene, not
image style or camera work.
Reply with JSON only:  {{``match'':  true or false, ``confidence'':
0.0--1.0, ``reason'':  ``<at most 20 words>''}}
```

User prompt for a subsequent transition state frame

```
You are shown two transition state frames from the same video.
Image 1 is the PREVIOUS transition state frame (reference only).
Image 2 is the CURRENT transition state frame {slot_pos} of
{total}.
Video context:  physical category = ``{label}''; caption =
``{caption}''
The annotation claims that by the CURRENT transition state frame
the scene has reached this state:  ``{state_text}''
Question:  comparing against image 1 where the claim describes
change or motion, does image 2 show that the scene has actually
reached the claimed state?  Judge the physical state of the scene,
not image style or camera work.
Reply with JSON only:  {{``match'':  true or false, ``confidence'':
0.0--1.0, ``reason'':  ``<at most 20 words>''}}
```

Table 12: **Results on PhyGenBench.** Model size, and different physical-consistency metrics are compared across multiple baseline methods, with **bold** and underlined values indicating the best and second-best scores, respectively. Our results are reported with text-to-video (T2V) setting.

Method	# param	Mechanics ↑	Optics ↑	Thermal ↑	Material ↑	Average ↑
Open-Sora v2 (Peng et al., 2025)	11B	55.00	66.67	47.78	50.83	56.25
LTX-Video (HaCohen et al., 2024)	13B	45.83	65.33	43.33	37.50	49.37
Cosmos3-Nano (NVIDIA, 2026)	16B	57.50	75.33	48.89	46.67	58.75
CausalMotion Zhuang et al. (2026)	13B	**73.33**	**75.33**	**65.56**	64.17	**70.21**
Ours (CogVideoX-2B-T2V)	20B+2B	60.00	73.33	63.33	**66.04**	65.68

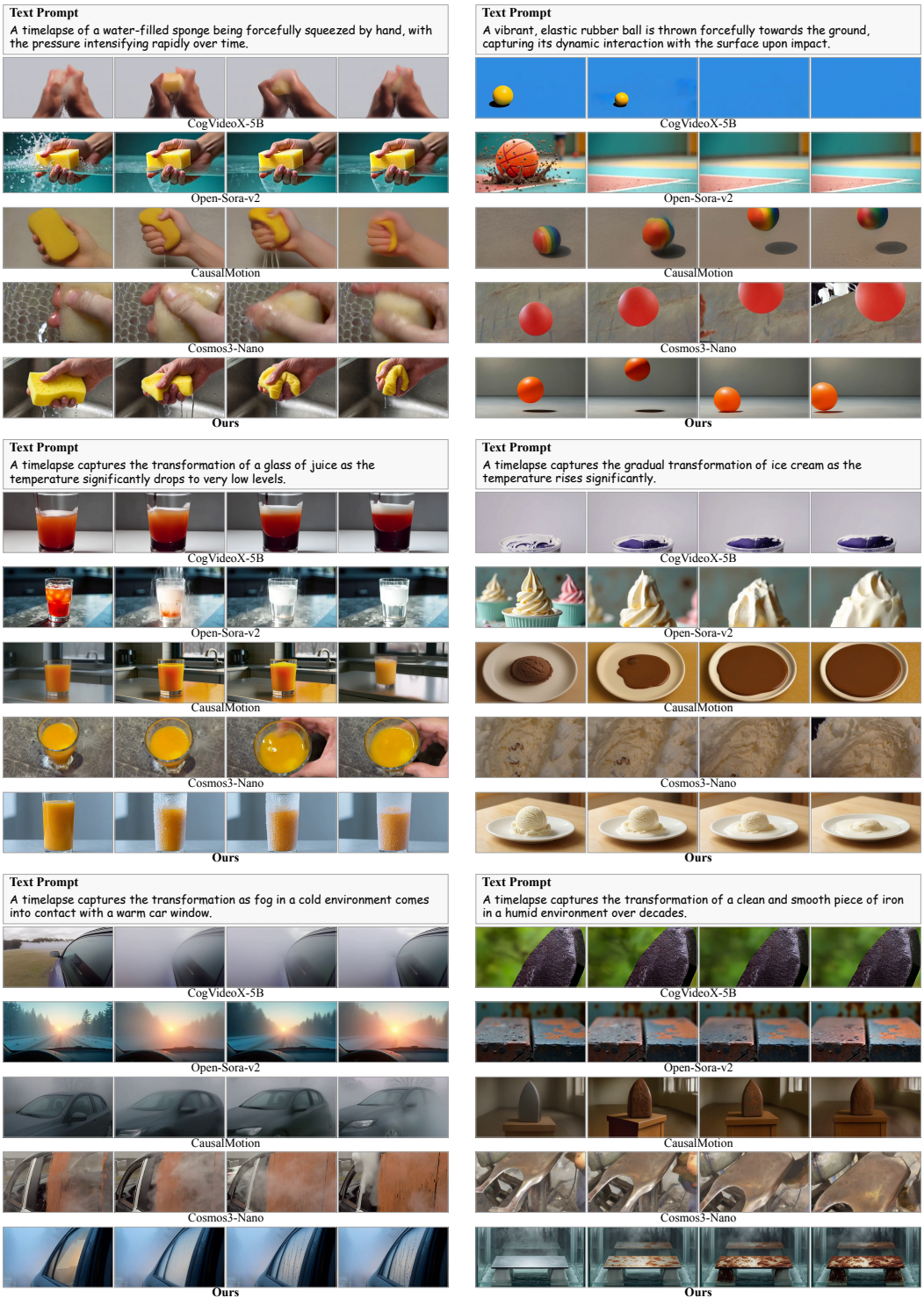


Figure 7: **Qualitative comparisons on PhyGenBench**. Each block shows the text prompt and four frames from CogVideoX-5B, Open-Sora-v2, CausalMotion, Cosmos3-Nano, and our method.