PHYSCHAIN: CONSTRAINT-BASED PHYSICAL CHAINS FOR VISUAL STATE AND TRANSITION GENERATION

Anonymous authors
Paper under double-blind review



Figure 1: We propose PHYSCHAIN, a framework that simulates a chain of future visual states and transitions under physical constraints to generate long-horizon physically valid videos. Our method enables the simulation of diverse real-world dynamics, such as cascade motions, complex mechanics, thermodynamics, fluid dynamics, and robotics.

ABSTRACT

While existing methods for video generation leveraging the physical reasoning of Vision-Language Models (VLMs) have made impressive progress, they focus on injecting the knowledge into video generation through designated representations (e.g., physical descriptions, property labels, or motion trajectories), bounding the dynamics they can express. In this work, we present PHYSCHAIN, a framework that simulates a chain of future states with VLMs to generate physically valid videos from a single image. Our key idea is to simulate the chain of states and transitions under individually verifiable physical constraints to tractably handle real-world complex dynamics, highly entangled and implicit. For consistent and reliable simulations, we design a predict-and-correct loop with sampling-based constraint derivation, and further introduce a novel confidence-adaptive rendering scheme that tolerates the prediction errors that still remain after the simulations. In the experiments, PHYSCHAIN consistently outperforms competing physics-aware methods across five physics benchmarks, ranking first on most of them even among large-scale video foundation models, and generalizes to complex real-world dynamics with various potential applications. Videos are available at our anonymous project page.

1 INTRODUCTION

Video Models are positioned as core components of applications such as world models and virtual environment generation, making physical validity a requirement beyond visual plausibility. To inject physical knowledge into video generation, Vision-Language Models (VLMs)' strong physical reasoning capability has mainly been leveraged in diverse representations such as binary preference judgments, physical descriptions, or property labels (Ji et al., 2025; Wang et al., 2025a) (Fig. 2a).

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.

However, the ambiguity of these representations limits the capture of real-world physics dynamics, which are highly implicit and complex. Beyond producing an intermediate representation as reasoners, CoECT (Wang et al., 2026c), VLIPP (Yang et al., 2025a), and DeforM (Li et al., 2026c) leverage VLMs to plan the future visual states[1], as *simulators*. They endow the VLMs with a specific representation (e.g., bounding boxes, drag) as a tool, enabling them to explicitly model physical dynamics based on physical formulations and reasoning (Fig. 2b). However, the dynamics that can be modeled are inherently limited to that representation, failing in reliable simulations.

We focus on how humans simulate future physical states to answer "what and how happens next." Rather than formulating state transitions according to physical laws, humans empirically learn a set of *constraints* from individual observable cues that reveal these transitions, and predict future states that satisfy these constraints. This mechanism motivates a tractable way to simulate entangled physical dynamics, without relying on the specific representation that such explicit modeling requires. Realizing it with VLMs, however, is not straightforward. Even when VLMs reason the constraints well, as Huang et al. (2025) and Wei et al. (2026) show for explicitly stated conditions, generation still fails to realize the states that satisfy them, which humans can imagine directly.

To address this, we introduce PHYSCHAIN, a framework that simulates a chain of future visual states with VLMs to generate physically valid videos from a single image. Rather than drawing the states at once, we design the simulation as a predict-and-correct loop. The loop runs over the entire sequence, which allows us to simulate a chain of physical actions faithfully by modeling the transitions over the whole chain of states without error accumulation in causal simulation (Yang et al., 2025a; Wang et al., 2026c). For consistent and reliable simulation, we carefully consider what the prediction draws and what the correction verifies. The prediction should draw from descriptions that reduce drawing to composition, as generation degrades once a prompt demands reasoning (Li et al., 2026b; Niu et al., 2026). For the correction, the constraints should be atomic, the form in which VLMs judge physical validity reliably (Cho et al., 2024; Meng et al., 2025), and should stay fixed throughout the simulation, as the judgments of VLMs drift across queries (Chen et al., 2024). We therefore let VLMs reason the dynamics from the input once and organize their observable cues, the individual visual facts in pixel space, into a structured memory. From this memory, we take a sampling-based approach to derive both the descriptions to predict the states and the constraints to verify them: an observable specifies a fact to be drawn, directly composed, and serves as a constraint asking whether that fact is observed, an individual reference shared throughout the simulation. The simulation proceeds iteratively, predicting the states from the descriptions, verifying both the states and the transitions between them against the constraints, and revising what the states depict and when the transitions occur. To render the simulated states into a video, we further introduce a confidence-adaptive video rendering scheme that tolerates the prediction errors remaining after the simulation, preserving the verified physics while refining the rest with the generative prior of the video model. Extensive experiments across five judgment-based and measurement-based benchmarks show that PHYSCHAIN consistently outperforms all competing physics-aware methods in overall performance, ranking first on most benchmarks even among large-scale video foundation models, and improves visually assessed physical plausibility, agreement with real-world trajectories, and adherence to Newtonian dynamics. We also demonstrate the applicability to complex physical chains, e.g. Rube Goldberg Machines.

Our contributions can be summarized as follows:

- We present PHYSCHAIN, a framework that plans a chain of future visual states and transitions under physical constraints in a predict-and-correct loop. Our constraint-based simulation makes entangled dynamics tractable, while remaining consistent and reliable.
- We introduce a confidence-adaptive rendering scheme that guides an off-the-shelf video model according to the confidence of each simulated state. The verified states are thereby preserved while the rest is refined under their guidance with the model's generative prior.
- We demonstrate the effectiveness of PHYSCHAIN through diverse experiments using both judgment-based and complementary measurement-based evaluations, showing significant improvements in visually assessed physical plausibility and quantitatively measured physical accuracy.

[1]By *visual states*, we refer to the state of the scene represented in pixel space, which is not a frame of the video itself but serves as a prior for rendering it.

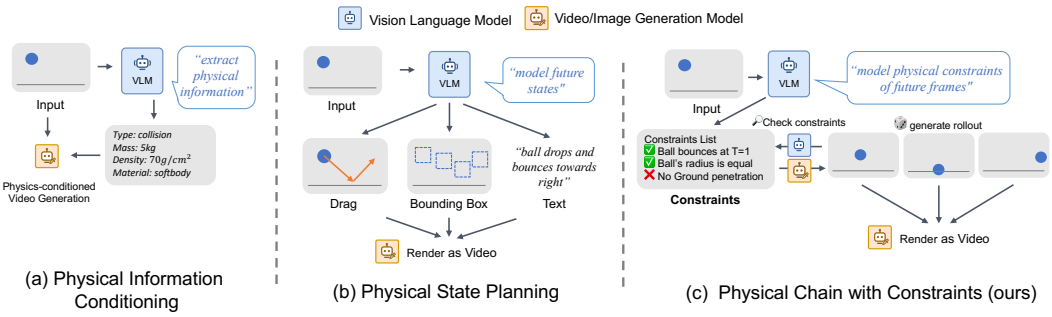

Figure 2: **How VLMs are leveraged for physically plausible video generation.** **(a)** They use VLM reasoning to extract and condition physical information, which remains ambiguous about the dynamics. **(b)** A set of methods use VLMs to explicitly plan the future states through a designated representation, limiting the dynamics they can express. **(c)** We instead let VLMs derive constraints and simulate a chain of the future states to satisfy them, which specify the complex dynamics without a designated representation.

2 Related Work

2.1 Physics-aware Video Generation

Recent work on video generation has moved beyond visual quality toward physical validity by incorporating physical knowledge from external sources. Liu et al. (2024); Chen et al. (2025a); Li et al. (2025c); Zhan et al. (2026); Liu et al. (2026); Foo et al. (2026); Kim et al. (2026); Zhang et al. (2026) draw it from physics solvers, reconstructing the input into geometry and material parameters, then rendering the simulated motion back into videos. Solvers yield exact dynamics, while the supported scope is bounded by the simulator in use and mostly object-centric scenarios. Another line of work leverages the physical reasoning of VLMs. Zhang et al. (2025); Ji et al. (2025); Cai et al. (2026); Chen et al. (2025b); Wang et al. (2025a); Pathak et al. (2026); Wang et al. (2026b); Cheng et al. (2026) turn it into a condition the video model is trained on, such as physical descriptions and property labels, or rewards and preferences distilled from VLM judgments, while VLIPP (Yang et al., 2025a), CoECT (Wang et al., 2026c), and DeforM (Li et al., 2026c) have VLMs explicitly model the dynamics as motion trajectories (e.g., bounding boxes or drags), which a video model renders. In contrast, PhysChain simulates future states under physical constraints derived by VLMs.

2.2 VLM Verification and Synthesis

A wide range of work adopts VLMs for verifying and synthesizing visual content. For verification, VLMs judge the physical validity of generated videos (Bansal et al., 2025; Meng et al., 2025; Lin et al., 2026). Their reliability depends on how the judgment is posed: a judgment on the whole video correlates poorly with human ratings, while questions where each covers a single semantic unit, restore the correlation (Hu et al., 2023; Cho et al., 2024; Meng et al., 2025). However, well-posed judgments remain unstable, shifting with how the question is rephrased, how the image is styled, and in which format the answer is asked (Zhang et al., 2024; Chen et al., 2024; Chou et al., 2025). Another line of work discusses asymmetry of encoding and decoding of models for physical scenes. They show that the reasoning of VLMs transfers poorly to generation. Huang et al. (2025); Wei et al. (2026) report that a condition stated explicitly in the prompt is often not realized, and even unified models answer in language what they fail to render in pixels (Li et al., 2026a; Wang et al., 2026a). Li et al. (2026b); Niu et al. (2026) nonetheless show prompts asking for composition faring far better than those demanding reasoning. We design our simulation around these properties, and our rendering tolerates what synthesis still fails to realize.

3 PhysChain

We first formulate the problem (Sec. 3.1), and then introduce our constraint-based visual state simulation, which derives physical constraints (Sec. 3.2) and simulates visual states under them (Sec. 3.3). We finally describe how the simulated states are rendered into videos (Sec. 3.4).

3.1 Problem Formulation

Our goal is to achieve physically plausible video generation from a single image. We attain this by VLM-driven constraint-based visual state simulation. Given a first frame $S_0 \in \mathbb{R}^{H\times W\times 3}$ representing the initial visual state and a text description y of the target video, our proposed PhysChain

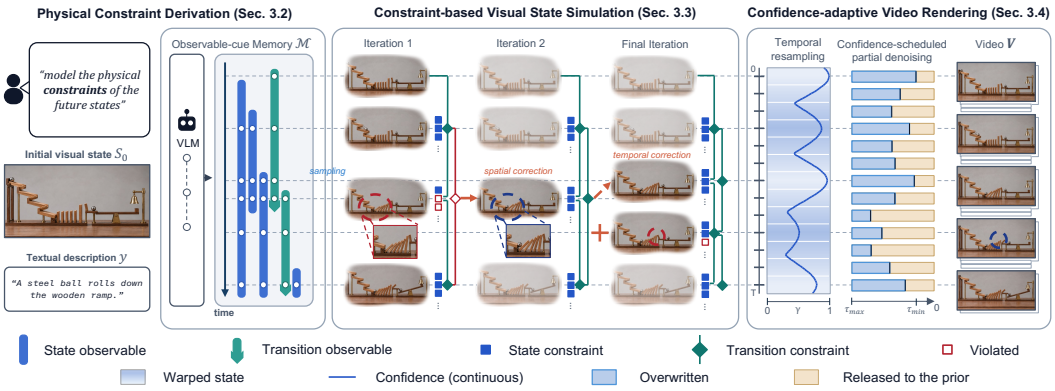

Figure 3: **Overview of PHYSCHAIN.** We simulate a series of future visual states and transitions under VLM-derived physical constraints through a predict-and-correct loop. Confidence-adaptive rendering then guides a video model to preserve verified states while refining the remaining predictions with its generative prior.

simulates the visual states $\{S_k\}_{k=1}^n$ to satisfy the physical constraints $\{\mathcal{C}_k\}_{k=1}^n$ derived from S_0 and y, and renders the simulated states into a video $\mathbf{V} \in \mathbb{R}^{(T+1)\times H\times W\times 3}$.

To enable reliable simulation and rendering, regarding the properties discussed in Sec. 2.2, we must address how to **(i)** make each physical constraint atomic enough for VLMs to verify reliably, while structuring the constraints to capture the entangled relations of the dynamics and keeping them as a consistent reference throughout the simulation; **(ii)** design the simulation around the poor transfer of VLM reasoning to synthesis, drawing each state by composing facts rather than inferring them; and **(iii)** tolerate at rendering the states that synthesis fails to realize after the simulation.

3.2 PHYSICAL CONSTRAINT DERIVATION

The core of physical constraints lies in serving as an individually verifiable reference for judging the validity of entangled physics dynamics. Yet generating them directly from the initial visual state and text description yields constraints reasoned independently, unstructured and inconsistent across the simulation, prone to redundancy and contradiction that undermine their reliability as a reference.

Observable-cue Memory. To address the above limitation, we build a structured memory $\mathcal{M}$ that organizes the observable cues of the reasoned physical dynamics, from which the constraints are derived. The memory is constructed once from S_0 and y as a graph representation through chain-of-thought (CoT) (Wei et al., 2022) reasoning (refer to Sec. B.1 of the supplementary for details), and serves as a consistent reference for deriving the constraints on the future states throughout the simulation. Note that we use the graph representation to retain the observable cues in a structured form, not to model the relations underlying the physical dynamics as in prior art (Wang et al., 2026c). Formally, the memory $\mathcal{M}$ consists of entities $\mathcal{E}$ such as objects, background, and visual phenomena, events $\mathcal{V}$ as the physical happenings among the entities, and observables $\mathcal{O}$ as the individual visual facts in pixel space that reveal each event, together with the connections among them. We distinguish two types of observables: state observables $\mathcal{O}^{\mathrm{S}}$, which describe visual facts that hold over a time span, and transition observables $\mathcal{O}^{\mathrm{T}}$, which describe changes that progress over time. To enable atomic observables that retain their dependencies, we further order them without cycle by presupposition[2] following the dependency graph of Cho et al. (2024), building each verification on those it presupposes.

Constraint Sampling. The observables $\mathcal{O}$ in the constructed memory $\mathcal{M}$ are visual facts by definition, hence each naturally acts as a constraint that asks whether the fact is observed in a given state. To derive constraints, we only need to sample the observables active at the target time instants from those distributed over the entire dynamics. Given n time instants $\{t_k\}_{k=1}^n$ of the future states, the constraints are derived as:

$$\{\mathcal{C}_k\}_{k=1}^n = \Pi(\mathcal{M}, \{t_k\}_{k=1}^n), \tag{1}$$

[2]An observable presupposing another holds only if the latter holds; for instance, whether the steel ball strikes the dominoes is verified only if the steel ball is present in the state.

where Π denotes the sampling that assigns the given time instants to the observables in $\mathcal{M}$ and compiles them into constraints. Each $\mathcal{C}_k$ is the set of constraints at t_k, consisting of state constraints $\mathcal{C}_k^{\mathrm{S}}$ sampled from $\mathcal{O}^{\mathrm{S}}$ whose span contains t_k and transition constraints $\mathcal{C}_k^{\mathrm{T}}$ sampled from $\mathcal{O}^{\mathrm{T}}$ whose span covers $\{t_{k-1}, t_k, t_{k+1}\}$ to verify the progression. This sampling-based constraint derivation makes our simulation flexible and consistent, as it enables us to freely select and change the time instants in both number and position while the constraints stay grounded in the same structured memory $\mathcal{M}$ without additional reasoning.

3.3 Constraint-based Visual State Simulation

We now describe how the visual states are simulated under the derived constraints. Since the reasoning of VLMs transfers poorly to synthesis, we design the simulation as a *predict-and-correct* loop, where the strong reasoning of VLMs corrects what generation fails to realize. Note that our simulation iterates over the entire sequence to avoid the error accumulated by simulating the states causally along time as in prior art (Yang et al., 2025a; Wang et al., 2026c).

Simulation Setup. We set two elements corrected during the simulation: the time instants $\{t_k\}_{k=1}^n$ of the future states for the *temporal* correction of when they occur, and their descriptions $\{d_k\}_{k=1}^n$ for the *spatial* correction of what they depict. The initial time instants are chosen to sample at least one constraint from each observable in $\mathcal{O}$. At each instant t_k, the initial description d_k is then written by composing the observables sampled at that instant. During the simulation, the time instants can be corrected in both their number and their positions, as long as a constraint is still sampled from each observable, while the descriptions $\{d_k\}_{k=1}^n$ are kept throughout with corrections appended on top.

Prediction. The prediction Φ draws the future visual states $\{S_k\}_{k=1}^n$ at the time instants $\{t_k\}_{k=1}^n$ from their descriptions $\{d_k\}_{k=1}^n$. Each state is generated from its description with an image editing model, referring to the initial visual state S_0 and the text description y of the target video:

$$\{S_k\}_{k=1}^n = \Phi\big(\{t_k\}_{k=1}^n, \{d_k\}_{k=1}^n \mid S_0, y\big). \tag{2}$$

Correction. The correction Ψ verifies the predicted states $\{S_k\}_{k=1}^n$ against the constraints $\{\mathcal{C}_k\}_{k=1}^n$ derived by Eq. 1 and revises the time instants and the descriptions:

$$\big(\{\tilde{t}_k\}_{k=1}^{\tilde{n}}, \{\tilde{d}_k\}_{k=1}^{\tilde{n}}\big) = \Psi\big(\{S_k\}_{k=1}^n, \{\mathcal{C}_k\}_{k=1}^n \mid S_0, y\big), \tag{3}$$

where $\{\tilde{t}_k\}_{k=1}^{\tilde{n}}$ and $\{\tilde{d}_k\}_{k=1}^{\tilde{n}}$ are the revised time instants and descriptions, whose number $\tilde{n}$ may differ from n as the correction adds time instants. The constraints are then re-sampled at $\{\tilde{t}_k\}_{k=1}^{\tilde{n}}$ by Eq. 1 and the states redrawn from $\{\tilde{d}_k\}_{k=1}^{\tilde{n}}$ by Eq. 2, so that prediction Φ and correction Ψ iterate until all the constraints are satisfied or a maximum number of iterations N_{iter} is reached. We now detail how the states are verified and how the two elements are revised from the verification.

Verification. The predicted states $\{S_k\}_{k=1}^n$ are presented to VLMs together with the constraints $\{\mathcal{C}_k\}_{k=1}^n$, and each constraint is judged on whether it is satisfied, with a reason required when it is not. To judge the validity of the dynamics across the states, they are presented at once with S_0 and y, as the transition constraints $\mathcal{C}_k^{\mathrm{T}}$ are defined over a state and the surrounding instants. The judgments are then propagated along the presupposition: when a constraint is violated, every constraint that presupposes it is also treated as violated, which keeps the verification consistent and lets the revision focus on the constraint the others depend on. From these judgments, we compute the confidence of each state as the fraction of its constraints that remain satisfied, which guides the following revision and the rendering (Sec. 3.4). Let $a_c \in \{0, 1\}$ denote whether a constraint c is satisfied, then the confidences are given as:

$$\{\gamma_k\}_{k=1}^n = \left\{ \frac{1}{|\mathcal{C}_k|} \sum_{c \in \mathcal{C}_k} a_c \right\}_{k=1}^n. \tag{4}$$

Revision. The violations do not indicate which states to revise on their own, since those of $\mathcal{C}_k^{\mathrm{T}}$ span several states at once. VLMs therefore read the violations and their reasons across the whole sequence and decide which states account for them. From this attribution, we obtain $\tilde{d}_k$ by augmenting the original description d_k of each state to be revised with its violated constraints. The time instants $\{t_k\}_{k=1}^n$ are corrected only where what the states depict is already valid, as their placement is judged from the states themselves. When the confidence of every such state reaches a threshold

γ_{th}, VLMs reason about whether the event is placed at the right time and simulated at the right density. Through this, we obtain $\{\tilde{t}_k\}_{k=1}^{\tilde{n}}$ by revising the time instants of each event accordingly.

3.4 Confidence-adaptive Video Rendering

We now present our rendering of the simulated visual states $\{S_k\}_{k=1}^{n}$[3] into the video $\mathbf{V}$. Although our simulation is carefully designed to be flexible and reliable, it still shares the inherent limitation of VLM-driven simulation, leaving the states only at the selected instants and subject to prediction errors. Following Yang et al. (2025a); Wang et al. (2026c), we leverage the generative prior of a video model as a renderer to refine the frames guided by the simulated states, while further adapting the guidance to their confidences $\{\gamma_k\}_{k=1}^{n}$ to tolerate the remaining errors.

Temporal Resampling. The states $\{S_k\}_{k=1}^{n}$ lie at the simulated instants $\{t_k\}_{k=1}^{n}$, while the video $\mathbf{V}$ consists of S_0 followed by T frames at the instants $\{t_m\}_{m=1}^{T}$ fixed by the video model. We therefore resample both the states and their confidences from $\{t_k\}_{k=1}^{n}$ to $\{t_m\}_{m=1}^{T}$. For the states, interpolating between them directly, as in prior art (Wang et al., 2026c), blends them into ghosting artifacts that misguide the video model. To address this, we warp each state in $\{S_k\}_{k=0}^{n}$ along its displacement to the next, obtained from dense correspondences (Feng et al., 2025). The confidences follow accordingly: a frame inherits the confidence interpolated between the two states it lies between, which further decays with its distance from them since the warped frames are no longer verified states. To model this decay, we place at each t_k a Gaussian kernel, which reflects the quadratic error of the first-order warp, and scale the interpolated confidence by the maximum kernel value at each t_m:

$$\{\bar{S}_m\}_{m=1}^{T} = \{\mathcal{W}(S_k, \lambda_m \mathbf{u}_k)\}_{m=1}^{T}, \qquad \{\bar{\gamma}_m\}_{m=1}^{T} = \Big\{\hat{\gamma}_m \max_j \exp\Big(-\frac{(t_m - t_j)^2}{2h^2}\Big)\Big\}_{m=1}^{T}, \tag{5}$$

where k satisfies $t_k \le t_m < t_{k+1}$ with $\lambda_m = (t_m - t_k)/(t_{k+1} - t_k)$, $\mathbf{u}_k$ is the displacement from S_k to S_{k+1}, $\mathcal{W}$ warps S_k by the given displacement, $\hat{\gamma}_m = (1-\lambda_m)\gamma_k + \lambda_m \gamma_{k+1}$ is the confidence interpolated between the two states, h is the bandwidth of the kernel, and $\gamma_0 = 1$ for the given S_0.

Confidence-scheduled Partial Denoising. To preserve the resampled states where reliable and refine the rest with the generative prior of the video model, we render the video by partially denoising the states from frame-wise noise levels set by their confidences, where a lower level preserves more of a state. We first encode the states into video latents $\bar{\mathbf{z}}$ consisting of L latent frames, and sample a noise $\boldsymbol{\epsilon}$ once to obtain the noised video latents $\bar{\mathbf{z}}_\tau = (1-\tau)\cdot\bar{\mathbf{z}} + \tau\cdot\boldsymbol{\epsilon}$ at any noise level τ. Since standard video models jointly denoise all latent frames at each denoising step, we realize the frame-wise noise levels by overwriting, where how long each latent frame is overwritten during denoising determines its actual starting noise level. We set the threshold for the ℓ-th of the L latent frames from its confidence as

$$\tau_{\text{th}}^{\ell} = \bar{\gamma}_{m_\ell} \cdot \tau_{\min} + (1 - \bar{\gamma}_{m_\ell}) \cdot \tau_{\max}, \tag{6}$$

where m_ℓ indexes the resampled frame corresponding to the ℓ-th latent frame, and $\tau_{\min}$ and $\tau_{\max}$ are hyperparameters bounding the range of the threshold. Starting from $\bar{\mathbf{z}}_\tau$, where τ initially set to the noise threshold of the least reliable latent frame, we overwrite the ℓ-th latent frame with its noised version $\bar{\mathbf{z}}_\tau^{\ell}$ at each denoising step until the level reaches its threshold, and leave it to the prior afterwards by the standard denoising. A reliable frame is thus overwritten longer to be preserved, while an unreliable one is released earlier to be refined by the prior under the guidance of the frames still being overwritten through the temporal attention of the video model. We detail the full procedure in Alg. 1. Note that Yang et al. (2025a); Wang et al. (2026c) inject noise at a single level regardless of the reliability of each frame, which inevitably erases the physics of the reliable frames along with the errors of the unreliable ones, whereas our scheme preserves the verified physics while refining the rest.

4 Experiments

4.1 Setup

Implementation Details. Unless otherwise stated, we use the following setting. For the simulation, we use Gemini 3.1 Pro (Google DeepMind, 2026) for the VLMs and Gemini 3 Pro Image (Google DeepMind, 2025a) for the prediction (*Ours-closed*), and additionally report an open-source variant with Qwen3-VL-8B (Bai et al., 2025) and Qwen-Image-2.1 (Qwen Team, 2026b)

[3] With a slight abuse of notation, we hereafter denote by $\{S_k\}_{k=1}^{n}$, $\{t_k\}_{k=1}^{n}$, and $\{\gamma_k\}_{k=1}^{n}$ the visual states, time instants, and confidences from the last iteration of the simulation.

(*Ours-open*), with $N_{\text{iter}} = 3$ and $\gamma_{\text{th}} = 0.85$ for both. For rendering, we use Wan2.2-5B (Wan Team, 2025) as our video model backbone, which is relatively computationally efficient, with $\tau_{\min} = 0.85$, $\tau_{\max} = 0.97$, and $h = 0.3$. We provide additional implementation details in Sec. B of the supplementary.

Evaluation Metrics & Datasets. We evaluate PHYSCHAIN on five benchmarks grouped by evaluation approach. **Measurement-based benchmarks** assess generated dynamics using numerical metrics: Physics-IQ Verified (Rädsch et al., 2026) measures spatiotemporal agreement with real recordings across five physical domains; PISA (Li et al., 2025a) measures free-fall trajectory errors against real recordings; and Morpheus (Tragoudaras et al., 2026) evaluates conservation laws and equations of motion in Newtonian rigid-body systems. **Judgement-based benchmarks** assess videos through visual judgments: PhyGenBench (Meng et al., 2025) evaluates physical phenomena across mechanics, optics, thermal, and material properties; and WorldModelBench (Li et al., 2025b) evaluates instruction following, common sense, and physical adherence across seven domains.

Baselines. We compare PHYSCHAIN against Wan2.2-5B (Wan Team, 2025), the video foundation model serving as our backbone, which delivers strong performance while remaining relatively lightweight. We include two groups of physics-aware methods built on video backbones of comparable scale, sharing our goal of improving the physical plausibility of video foundation models via VLMs. **Physical information conditioning methods** supply physical information as a condition to the video model: WISA (Wang et al., 2025a), the representative method conditioning on physical descriptions and property labels, and PhysRAG (Cheng et al., 2026), the most recent one retrieving physical knowledge. **Physical state planning methods** have VLMs explicitly model future states through designated representations: VLIPP (Yang et al., 2025a), CoECT (Wang et al., 2026c), and DeforM (Li et al., 2026c), whose VLMs are replaced with ours for fair comparison. In Sec. A.4, we further report the results of large-scale video foundation models such as Cosmos 3 (NVIDIA, 2026) and MiniMax H3 (MiniMax, 2026), as well as other physics-aware methods (Chen et al., 2025b; Cai et al., 2026), taken from their official leaderboards and papers for reference. For text-to-video (T2V) generation, image-to-video (I2V) methods are initialized with the same first frame, generated from the text prompt using Gemini 3 Pro Image (Google DeepMind, 2025a).

4.2 RESULTS

We present qualitative results of PHYSCHAIN and quantitative comparisons with baselines across five benchmarks. In Sec. A of the supplementary, we provide applications of PHYSCHAIN, qualitative comparisons with baselines, full evaluation results, and evaluations using alternative VLM and image-editing backbones. Generated videos for figures in the main paper and supplementary material are available on the anonymous project page.

PhyGenBench. PhyGenBench evaluates generated video across four physical domains: mechanics, optics, thermal and material, using open and closed VLMs. Note that the VLMs judging the judgement-based benchmarks (e.g., GPT-4o (OpenAI, 2024a), LLaVA-Interleave-DPO-7B (Li et al., 2024), InternVideo2 (Wang et al., 2024), and VQAScore (Lin et al., 2024) with CLIP-FlanT5-XXL in PhyGenBench) are disjoint from the VLMs used in our simulation, and the measurement-based benchmarks further assess the dynamics against real recordings and physical laws without any VLM judgment. PHYSCHAIN outperforms baselines and achieves the highest overall score of 0.75 (Fig. 4). This broad performance supports directly predicting future visual states, without restricting dynamics to designated intermediate representations such as bounding boxes or drag signals.

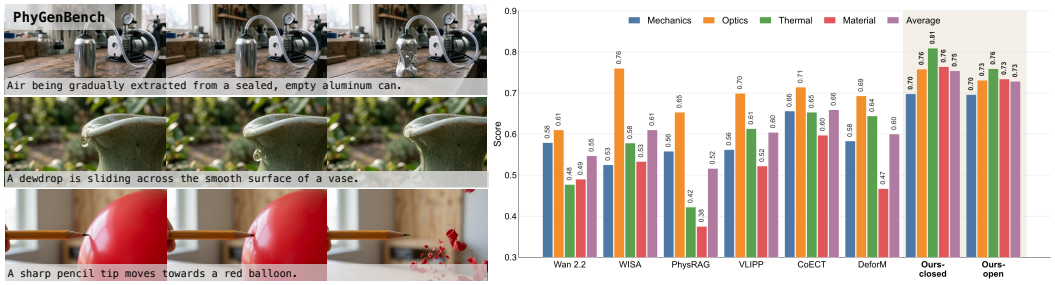

Figure 4: **Results on PhyGenBench. Left:** Generated sequences showing can deformation, droplet motion, and balloon puncture. **Right:** Overall scores, with PHYSCHAIN achieving the highest score.

WorldModelBench. WorldModelBench evaluates instruction following, common sense, and physical adherence using designated VLMs across diverse contexts, including animation, human

activities, animals, and games. PHYSCHAIN achieves the highest aggregate score among the compared methods (Fig. 5), highlighting its potential as a physically grounded world model that generates instruction-aligned future states across diverse scenarios.

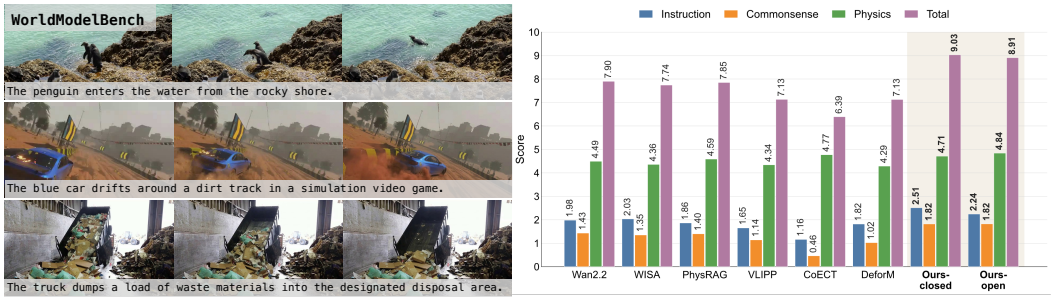

Figure 5: **Results on WorldModelBench. Left:** Generated sequences spanning animal motion, vehicle dynamics, and material dumping. **Right:** Scores for instruction following, common sense, and physics, with PHYSCHAIN achieving the highest aggregate score.

Physics-IQ Verified. Physics-IQ Verified compares generated videos with ground-truth recordings of controlled physical experiments using spatial and temporal overlap metrics and motion-based errors. It covers mechanics, including solid and fluid dynamics, as well as thermodynamics, optics, and magnetism, providing a broad test of physical prediction against observed outcomes. Fig. 6 presents generated sequences of cascading physical interactions alongside quantitative results. Notably, the score improves from 27.56 for our Wan2.2-5B backbone, a relatively lightweight video model, to 40.53 with PHYSCHAIN, demonstrating the effectiveness of our framework.

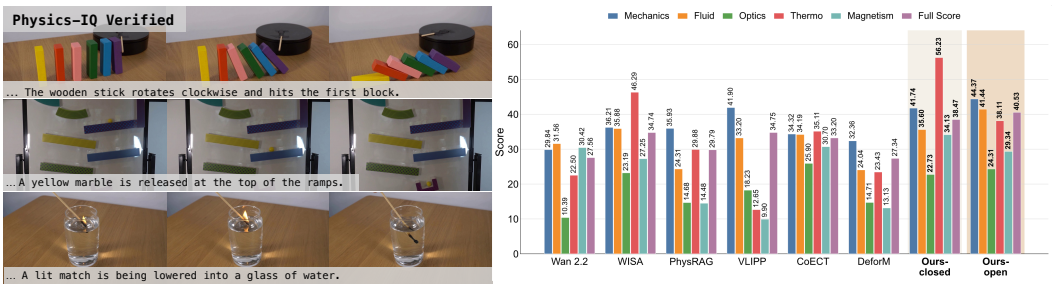

Figure 6: **Results on Physics-IQ Verified. Left:** Generated sequences showing domino collisions, a marble descending ramps, and a lit match entering water. **Right:** Quantitative comparison, with PHYSCHAIN achieving the highest score among the compared I2V baselines.

PISA. PHYSCHAIN(closed) achieves the lowest L2 and Chamfer distances (Fig. 7). This improved trajectory accuracy is consistent with our predict-and-correct loop, which verifies visual states and their transitions against physical constraints before rendering.

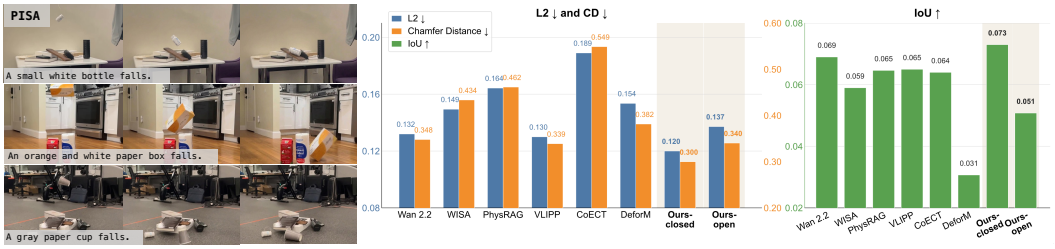

Figure 7: **Results on PISA Dropping. Left:** Generated sequences of falling objects. **Right:** Quantitative comparison using L2 and Chamfer distance (lower is better) and IoU (higher is better). PHYSCHAIN achieves the lowest errors and highest IoU among the compared methods.

Morpheus. Morpheus allows physically valid trajectories that differ from the recorded reference, and PHYSCHAIN achieves the highest score among the compared methods (Fig. 8), providing quantitative evidence of physical validity beyond visually assessed plausibility.

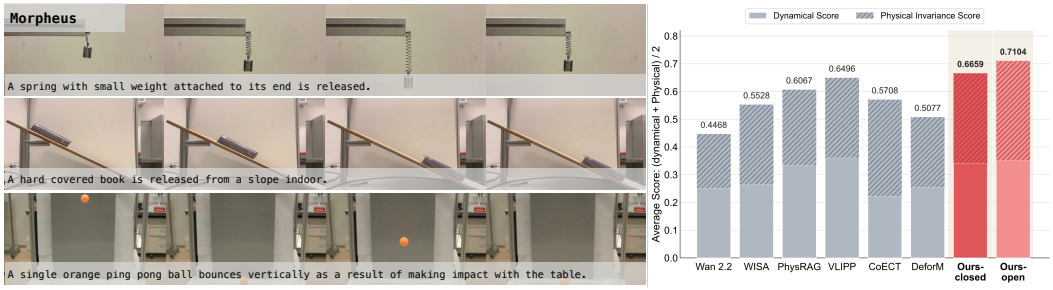

Figure 8: **Results on Morpheus. Left:** Generated sequences showing spring oscillation, sliding, and bouncing. **Right:** Newtonian physics scores, with PHYSCHAIN outperforming baselines.

4.3 ABLATION STUDY

Ablation on confidence-adaptive video rendering. In Tab. 1a, we report results without confidence-adaptive rendering on PhyGenBench, where a uniform denoising threshold of $\tau_{\text{th}} = 0.91$ is applied to all frames. The score decreases from 0.75 to 0.71, showing that adjusting the rendering process based on confidence improves physical consistency. Fig. 9 provides an example where confidence-adaptive video rendering corrects an abnormal state from visual state simulation.



Figure 9: **Qualitative example.** Confidence-adaptive rendering corrects a low-confidence state where the statue suddenly stands up.

(a) Component ablation	
Configuration	**Score ↑**
w/o Observable-cue Memory	0.64
w/o Adaptive Rendering	0.71
Ours (full)	**0.75**

(b) Iteration ablation	
N_{iter}	**Score ↑**
1	0.64
2	0.73
3 (default)	**0.75**
4	0.75
5	0.74

Table 1: **Ablation studies on PhyGenBench. Left:** Both observable-cue memory and confidence-adaptive rendering individually improve performance. **Right:** Performance saturates at three correction iterations, our default setting.

Ablation on observable-cue memory. In Tab. 1a, we ablate the observable-cue memory on PhyGenBench, with constraints derived directly at each instant while retaining the same CoT reasoning (see Sec. B.1 of the supplementary for details), reducing the score. Without the shared memory, independently derived constraints are less consistent and atomic across states and iterations, highlighting the importance of a persistent, structured reference throughout the simulation.

Ablation on number of iterations. In Tab. 1b, we vary the maximum number of predict-and-correct iterations N_{iter} from 1 to 5. With $N_{\text{iter}} = 1$, the states are predicted once from the initial descriptions derived from the observable-cue memory and verified only to compute their confidences for rendering, without any correction. Performance improves over the first three iterations as the corrections refine both what the states depict and when the transitions occur, then saturates as further corrections are not fully realized by the image editing model. We therefore adopt $N_{\text{iter}} = 3$ by default, balancing physical accuracy and computational cost.

5 CONCLUSION

We introduced PHYSCHAIN, a framework for generating physically valid videos by simulating chains of future visual states and transitions under physical constraints, making entangled physical dynamics tractable without restricting them to a designated representation. We design the simulation as a predict-and-correct loop over the entire chain, using shared observable-cue memory to consistently verify and refine states and transitions. We further introduce confidence-adaptive rendering to preserve reliable simulated states while refining unreliable ones with the generative prior of the video model. Extensive experiments demonstrate improved physical plausibility and accuracy, with further applications highlighting the versatility of our constraint-based visual state simulation. Future work will extend physical chains to longer horizons with real-time performance.

AI USE STATEMENT

In this work, we used generative AI tools for two purposes. First, we used a closed source large language model to edit the manuscript for grammar, clarity, and readability; all scientific content, claims, and the structure of the paper were written by the authors. Second, we used a closed source AI coding assistant to implement part of the source code, such as data loading, evaluation scripts, and utility functions; the core algorithm of PHYSCHAIN was designed and implemented by the authors, and all AI-assisted code was reviewed, tested, and verified for correctness by the authors.

We have not used generative AI tools to generate synthetic datasets, formulate hypotheses or mathematical claims, design the methodology or experiments, or interpret results. Note that Vision-Language Models (Gemini 3.1 Pro and Gemini 3 Pro Image) are used as components of the proposed method, as described in Sec. 3; this constitutes the subject of our research rather than AI assistance in the research process.

ETHICS STATEMENT

This work does not involve human subjects, personal data, or newly collected datasets; all experiments are conducted on publicly available benchmarks and their released prompts and images. As with any video generation method, PHYSCHAIN could be misused to produce misleading or fabricated visual content. Our method is designed to improve the physical validity of generated videos and does not add capabilities for depicting specific individuals or events beyond those of the underlying foundation models; we recommend that any deployment adopt the usage policies and safeguards of these models. The VLMs and image generators used in our framework may inherit biases from their training data, which can propagate into the generated videos. We have read and adhere to the ICLR Code of Ethics.

REPRODUCIBILITY STATEMENT

We provide all details necessary to reproduce our results. The full pipeline is described in Sec. 3 and 4, and all hyperparameters (e.g., the confidence threshold range for adaptive rendering) are reported there. All benchmarks (PhyGenBench, WorldModelBench, and the measurement-based benchmarks in Sec. 4) are publicly available, and we follow their official evaluation protocols. Since our method relies on proprietary API models, outputs may vary across API updates; to facilitate reproducibility, we additionally report Ours-open, which runs entirely on open-source backbones. We report the model versions and dates used, and we will release our code, prompts, and generated videos upon publication.

REFERENCES

Alibaba Cloud. Alibaba unveils wan2.6 series enabling everyone to star in videos. https://www.alibabacloud.com/en/press-room/alibaba-unveils-wan2-6-series-enabling-everyone, December 2025. Accessed: 2026-09-25.

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: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 *ICLR*, 2025.

Yuanhao Cai, Kunpeng Li, Menglin Jia, Jialiang Wang, Junzhe Sun, Feng Liang, Weifeng Chen, Felix Juefei-Xu, Chu Wang, Ali Thabet, Xiaoliang Dai, Xuan Ju, Alan Yuille, and Ji Hou. Phygdpo: Physics-aware groupwise direct preference optimization for physically consistent text-to-video generation. In *ECCV*, 2026.

Boyuan Chen, Hanxiao Jiang, Shaowei Liu, Saurabh Gupta, Yunzhu Li, Hao Zhao, and Shenlong Wang. Physgen3d: Crafting a miniature interactive world from a single image. *CVPR*, 2025a.

Dongping Chen, Ruoxi Chen, Shilin Zhang, Yaochen Wang, Yinuo Liu, Huichi Zhou, Qihui Zhang, Yao Wan, Pan Zhou, and Lichao Sun. MLLM-as-a-judge: Assessing multimodal LLM-as-a-judge with vision-language benchmark. In *ICML*, 2024.

Harold Haodong Chen, Haojian Huang, Qifeng Chen, Harry Yang, and Ser Nam Lim. Hierarchical fine-grained preference optimization for physically plausible video generation. *NeurIPS*, 2025b.

Kexu Cheng, Zicheng Liu, Mingju Gao, Chunhe Song, and Hao Tang. Physrag: Enhancing physics-awareness in video generation via retrieval-augmented generation. *ECCV*, 2026.

Jaemin Cho, Yushi Hu, Roopal Garg, Peter Anderson, Ranjay Krishna, Jason Baldridge, Mohit Bansal, Jordi Pont-Tuset, and Su Wang. Davidsonian scene graph: Improving reliability in fine-grained evaluation for text-to-image generation. In *ICLR*, 2024.

Shih-Han Chou, Shivam Chandhok, James J. Little, and Leonid Sigal. MM-R^3: On (in-)consistency of vision-language models (VLMs). In *Findings of the Association for Computational Linguistics: ACL 2025*, pp. 4762–4788, 2025.

fal.ai. Minimax h3 max: Image-to-video. https://fal.ai/models/minimax/h3-max/image-to-video, 2026. Post-trained variant of MiniMax H3.

Weichen Fan, Chenyang Si, Junhao Song, Zhenyu Yang, Yinan He, Long Zhuo, Ziqi Huang, Ziyue Dong, Jingwen He, Dongwei Pan, et al. Vchitect-2.0: Parallel transformer for scaling up video diffusion models. *arXiv:2501.08453*, 2025.

Haiwen Feng, Junyi Zhang, Qianqian Wang, Yufei Ye, Pengcheng Yu, Michael J Black, Trevor Darrell, and Angjoo Kanazawa. St4rtrack: Simultaneous 4d reconstruction and tracking in the world. In *ICCV*, 2025.

Lin Geng Foo, Mark He Huang, Alexandros Lattas, Stylianos Moschoglou, Thabo Beeler, and Christian Theobalt. Physical simulator in-the-loop video generation. In *CVPR*, 2026.

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. *NeurIPS*, 2025.

Nate Gillman, Yinghua Zhou, Zitian Tang, Evan Luo, Arjan Chakravarthy, Daksh Aggarwal, Michael Freeman, Charles Herrmann, and Chen Sun. Goal force: Teaching video models to accomplish physics-conditioned goals. *CVPR*, 2026.

Google DeepMind. Nano banana pro (gemini 3 pro image). https://deepmind.google/models/gemini-image/pro/, 2025a. Accessed: 2026-09-13.

Google DeepMind. Veo 3. https://deepmind.google/models/veo/, 2025b.

Google DeepMind. Gemini 3.1 pro. https://deepmind.google/models/model-cards/gemini-3-1-pro/, 2026. Accessed: 2026-09-13.

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:2501.00103*, 2024.

Yushi Hu, Benlin Liu, Jungo Kasai, Yizhong Wang, Mari Ostendorf, Ranjay Krishna, and Noah A. Smith. TIFA: Accurate and interpretable text-to-image faithfulness evaluation with question answering. In *ICCV*, 2023.

Kaiyi Huang, Chengqi Duan, Kaiyue Sun, Enze Xie, Zhenguo Li, and Xihui Liu. T2i-compbench++: An enhanced and comprehensive benchmark for compositional text-to-image generation. *TPAMI*, 2025.

Sihui Ji, Xi Chen, Xin Tao, Pengfei Wan, and Hengshuang Zhao. PhysMaster: Mastering Physical Representation for Video Generation via Reinforcement Learning. *arXiv:2510.13809*, 2025.

Yang Jin, Zhicheng Sun, Ningyuan Li, Kun Xu, Kun Xu, Hao Jiang, Nan Zhuang, Quzhe Huang, Yang Song, Yadong Mu, and Zhouchen Lin. Pyramidal flow matching for efficient video generative modeling. In *ICLR*, 2025.

Kandinsky Lab. Kandinsky wm 1.0: A family of models for physical ai. https://github.com/kandinskylab/kandinsky-wm, 2026.

Hwidong Kim, Yunho Kim, and Tae-Kyun Kim. 3dphysvideo: Consistency-guided flow sde for video generation via 3d scene reconstruction and physical simulation. *arXiv:2605.16795*, 2026.

Kuaishou. Kling AI. https://klingai.com/, 2025. Accessed: 2026-09-15.

Kuaishou Technology. Kling ai launches 2.5 turbo video model: Industry-leading performance, now more affordable. https://ir.kuaishou.com/news-releases/news-release-details/kling-ai-launches-25-turbo-video-model-industry-leading, September 2025. Accessed: 2026-09-25.

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. *ICML*, 2025a.

Dacheng Li, Yunhao Fang, Yukang Chen, Shuo Yang, Shiyi Cao, Justin Wong, Michael Luo, Xiaolong Wang, Hongxu Yin, Joseph E. Gonzalez, Ion Stoica, Song Han, and Yao Lu. Worldmodelbench: Judging video generation models as world models. In *CVPR Workshop*, 2025b.

Feng Li, Renrui Zhang, Hao Zhang, Yuanhan Zhang, Bo Li, Wei Li, Zejun Ma, and Chunyuan Li. LLaVA-NeXT-Interleave: Tackling multi-image, video, and 3D in large multimodal models. *arXiv:2407.07895*, 2024.

Hongxiang Li, Yaowei Li, Bin Lin, Yuwei Niu, Yuhang Yang, Xiaoshuang Huang, Jiayin Cai, Xiaolong Jiang, Yao Hu, and Long Chen. Gir-bench: Versatile benchmark for generating images with reasoning. In *ICLR*, 2026a.

Ouxiang Li, Yuan Wang, Xinting Hu, Huijuan Huang, Rui Chen, Jiarong Ou, Xin Tao, Pengfei Wan, Xiaojuan Qi, and Fuli Feng. Easier painting than thinking: Can text-to-image models set the stage, but not direct the play? In *ICLR*, 2026b.

Yunyi Li, Yu Qiao, Yaohui Wang, and Xinyuan Chen. Deform: Reasoning-guided physics-aware video generation via spatial-temporal masking. *ECCV*, 2026c.

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 *ICCV*, 2025c.

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:2605.10806*, 2026.

Zhiqiu Lin, Deepak Pathak, Baiqi Li, Jiayao Li, Xide Xia, Graham Neubig, Pengchuan Zhang, and Deva Ramanan. Evaluating text-to-visual generation with image-to-text generation. In *ECCV*, 2024.

Shaowei Liu, Zhongzheng Ren, Saurabh Gupta, and Shenlong Wang. Physgen: Rigid-body physics-grounded image-to-video generation. In *ECCV*, 2024.

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

Luma AI. Luma dream machine: Ai video generator. https://lumalabs.ai/dream-machine, 2024. Accessed: 2026-09-25.

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 *ICML*, 2025.

MiniMax. MiniMax H3: An open model breaking the boundaries between tasks and modalities. https://github.com/MiniMax-AI/MiniMax-H3, 2026. Accessed: 2026-09-23.

Saman Motamed, William Harvey, Benjamin Klein, Luc Van Gool, Zhuoning Yuan, and Ta-Ying Cheng. Void: Video object and interaction deletion. *ECCV*, 2026.

Yuwei Niu, Munan Ning, Mengren Zheng, Weiyang Jin, Bin Lin, Peng Jin, Jiaqi Liao, Chaoran Feng, Fanqing Meng, Kun-Peng Ning, Bin Zhu, and Li Yuan. WISE: World knowledge-informed semantic evaluation for text-to-image generation. In *ICML*, 2026.

NVIDIA. Cosmos-predict2: World simulation model for physical ai. https://research.nvidia.com/labs/cosmos-lab/cosmos-predict2/, 2025.

NVIDIA. Cosmos 3: Omnimodal world models for physical ai. https://research.nvidia.com/labs/cosmos-lab/cosmos3/technical-report.pdf, 2026. Accessed: 2026-09-23.

NVIDIA, Niket Agarwal, Arslan Ali, Maciej Bala, Yogesh Balaji, Erik Barker, Tiffany Cai, Prithvijit Chattopadhyay, Yongxin Chen, Yin Cui, Yifan Ding, Daniel Dworakowski, Jiaojiao Fan, Michele Fenzi, Francesco Ferroni, Sanja Fidler, Dieter Fox, Songwei Ge, Yunhao Ge, Jinwei Gu, Siddharth Gururani, et al. Cosmos world foundation model platform for physical ai. *arXiv preprint arXiv:2501.03575*, 2025.

OpenAI. GPT-4o system card. *arXiv:2410.21276*, 2024a.

OpenAI. Video generation models as world simulators. https://openai.com/index/video-generation-models-as-world-simulators/, 2024b. Accessed: 2026-09-25.

OpenAI. Sora 2 is here. https://openai.com/index/sora-2/, 2025.

Saurabh Pathak, Elahe Arani, Mykola Pechenizkiy, and Bahram Zonooz. Physvid: Physics aware local conditioning for generative video models. In *CVPR*, 2026.

Physics-IQ Verified. Physics-iq verified: I2v verified ranking. https://physics-iq-verified.anates.ai/, 2026. Accessed: 2026-09-25.

Pika Labs. Pika 2.0. https://pika.art/, 2024. Accessed: 2026-09-15.

Qwen Team. Qwen-robotworld technical report: Unifying embodied world modeling through language-conditioned video generation. *arXiv:2606.17030*, 2026a.

Qwen Team. Qwen-image-2.1: Compact, efficient, and unified image creation. https://qwen.ai/blog?id=qwen-image-2.1, sep 2026b. Accessed: 2026-09-23.

John Rachwan, Lennard Schaub, Nils Fleischmann, Johanna Sommer, Begüm Çig, Simon Langrieger, Amine Saboni, Gabriel Trégoat, Katalina Ortuno, and Quentin Sinig. Introducing P-Video: Fast and cheap video generation. https://www.pruna.ai/blog/p-video-fast-cheap-draft, February 2026.

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

Runway ML. Introducing gen-3 alpha. https://runway.com/research/introducing-gen-3-alpha, 2024. Accessed: 2026-09-25.

Tencent Hunyuan Foundation Model Team. Hunyuanvideo 1.5 technical report, 2025. URL https://arxiv.org/abs/2511.18870.

Antonios Tragoudaras, Chenyu Zhang, Daniil Cherniavskii, Antonios Vozikis, Thijmen Nijdam, Derck W. E. Prinzhorn, Mark Bodracska, Nicu Sebe, Andrii Zadaianchuk, and Stratis Gavves. Evaluating newtonian mechanics in video generative models with real physical systems. In *ICML*, 2026.

Wan Team. Wan2.2. https://github.com/Wan-Video/Wan2.2, 2025. Accessed: 2026-09-13.

Wan Team, 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:2503.20314*, 2025.

Chenlong Wang, Yuhang Chen, Zhihan Hu, Dongping Chen, Wenhu Chen, Sarah Wiegreffe, and Tianyi Zhou. Quantifying the gap between understanding and generation within unified multimodal models. In *CVPR Findings*, 2026a.

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. *NeurIPS*, 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. *IJCV*, 2025b.

Yi Wang, Kunchang Li, Xinhao Li, Jiashuo Yu, Yinan He, Guo Chen, Baoqi Pei, Rongkun Zheng, Zun Wang, Yansong Shi, Tianxiang Jiang, Songze Li, Jilan Xu, Hongjie Zhang, Yifei Huang, Yu Qiao, Yali Wang, and Limin Wang. InternVideo2: Scaling foundation models for multimodal video understanding. In *ECCV*, 2024.

Zijun Wang, Panwen Hu, Jing Wang, Terry Jingchen Zhang, Yuhao Cheng, Long Chen, Yiqiang Yan, Zutao Jiang, Hanhui Li, and Xiaodan Liang. Prophy: Progressive physical alignment for dynamic world simulation. *CVPR*, 2026b.

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 *CVPR*, 2026c.

Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, Brian Ichter, Fei Xia, Ed Chi, Quoc V Le, and Denny Zhou. Chain-of-thought prompting elicits reasoning in large language models. *NeurIPS*, 2022.

Xinyu Wei, Jinrui Zhang, Zeqing Wang, Hongyang Wei, Zhen Guo, Bairui Li, and Lei Zhang. TIIF-Bench: How does your T2I model follow your instructions? In *ECCV*, 2026.

xAI. Grok imagine video. https://x.ai/news/grok-imagine-api, 2026.

Jinbo Xing, Menghan Xia, Yong Zhang, Haoxin Chen, Wangbo Yu, Hanyuan Liu, Xintao Wang, Tien-Tsin Wong, and Ying Shan. Dynamicrafter: Animating open-domain images with video diffusion priors. *arXiv preprint arXiv:2310.12190*, 2023.

Xindi Yang, Baolu Li, Yiming Zhang, Zhenfei Yin, Lei Bai, Liqian Ma, Zhiyong Wang, Jianfei Cai, Tien-Tsin Wong, Huchuan Lu, et al. Vlipp: Towards physically plausible video generation with vision and language informed physical prior. In *ICCV*, 2025a.

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 *ICLR*, 2025b.

Jiahao Zhan, Zizhang Li, Hong-Xing Yu, and Jiajun Wu. Perpetualwonder: Long-horizon action-conditioned 4d scene generation. In *CVPR*, 2026.

Ke Zhang, Cihan Xiao, Jiacong Xu, Yiqun Mei, and Vishal M. Patel. Think before you diffuse: Infusing physical rules into video diffusion. *arXiv:2505.21653*, 2025.

Xin Zhang, Yabo Chen, Yijie Fang, Wanying Qu, Haibin Huang, Chi Zhang, Feng Xu, and Xuelong Li. PhysOmni: Physics-grounded multi-object scene generation from a single image with real-time interaction. In *ACM MM*, 2026.

Yuan Zhang, Fei Xiao, Tao Huang, Chun-Kai Fan, Hongyuan Dong, Jiawen Li, Jiacong Wang, Kuan Cheng, Shanghang Zhang, and Haoyuan Guo. Unveiling the tapestry of consistency in large vision-language models. In *NeurIPS*, 2024.

Zangwei Zheng, Xiangyu Peng, Tianji Yang, Chenhui Shen, Shenggui Li, Hongxin Liu, Yukun Zhou, Tianyi Li, and Yang You. Open-sora: Democratizing efficient video production for all. *arXiv:2412.20404*, 2024.

A Additional Results

A.1 Applications

Counterfactual Physical Prompting. Following VOID (Motamed et al., 2026), we investigate counterfactual video generation under object removal and its physical consequences. In Fig. 10, we remove the middle three dominoes from a domino chain toppled by a rotating arm, using the same removal mask for both methods. The mask is expanded into interaction-aware quadmasks using Gemini 3.1 Pro for VOID and used to guide VLM reasoning and subsequent visual state simulation in PhysChain. Our method generates a physically consistent counterfactual sequence in which the arm topples the blue and purple dominoes, while the yellow one remains upright across the gap.

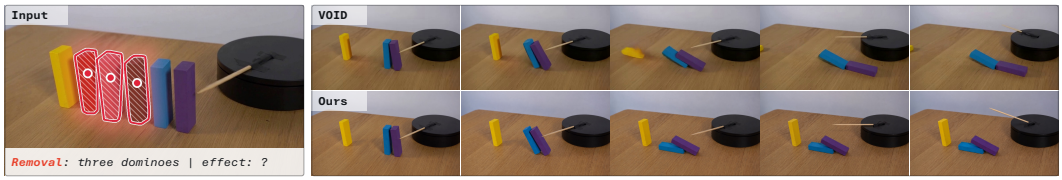

Figure 10: **Counterfactual physical prompting.** Removing the three middle dominoes (indicated by the red mask) disrupts the chain reaction, preventing the dominoes beyond the gap from toppling.

Quantitative Physical Prompting. Following Force Prompting (Gillman et al., 2025), we investigate video generation conditioned on an external force's magnitude, direction, and application point. In Fig. 11, we apply a backward force to the pink domino, specifying the same application point and direction for both methods. We set Force Prompting to its maximum normalized magnitude and specify 2 N for PhysChain through an annotated input image. Our results demonstrate quantitative physical prompting, where explicit numerical physical information is reflected in the generated dynamics, which video foundation models often struggle to follow through text prompts alone.

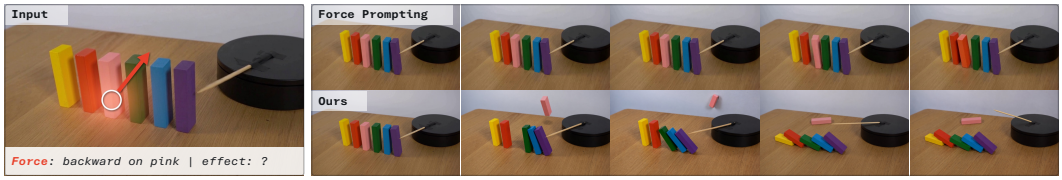

Figure 11: **Quantitative physical prompting.** Applying a backward force to the pink domino (indicated by the red arrow) induces physical interactions consistent with the specified force.

Inverse Physical Prompting. Following Goal Force (Gillman et al., 2026), we investigate video generation conditioned on a desired physical outcome without prescribing the initiating action. In Fig. 12, we set the same goal for both methods: toppling the yellow domino to the left. We specify the target through Goal Force's goal-force channel, leaving its direct-force and mass-conditioning channels unconditioned. For PhysChain, we provide only the goal-annotated image and an instruction to determine how to achieve it, without additional scene-specific descriptions. Our method generates a causal sequence of physical interactions, demonstrating its ability to infer and simulate the preceding events required to achieve the goal.

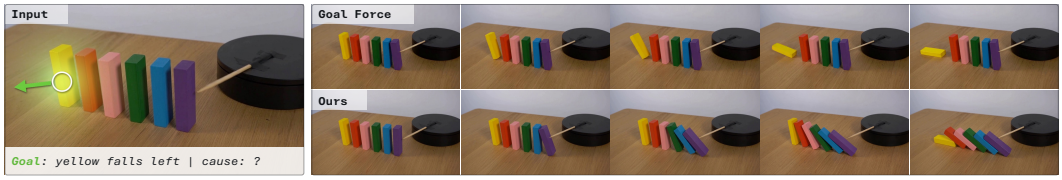

Figure 12: **Inverse physical prompting.** Specifying the yellow domino's leftward toppling as the goal (green arrow) guides preceding physical interactions without prescribing the initiating action.

A.2 Additional Qualitative Results

Fig. 13 provides a side-by-side comparison with planning-based methods. As VLIPP, CoECT, DeforM use VLMs to generate bounding box trajectory, drag signals, and spatial-temporal masks, VLM's physical knowledge does not fully transfer to the final video. Examples show that their intermediate representations limit their domain diversity, whereas PhysChain directly simulates in pixel space where all details are available, showcasing diverse scenarios, including liquid freezing and a chandelier shining.

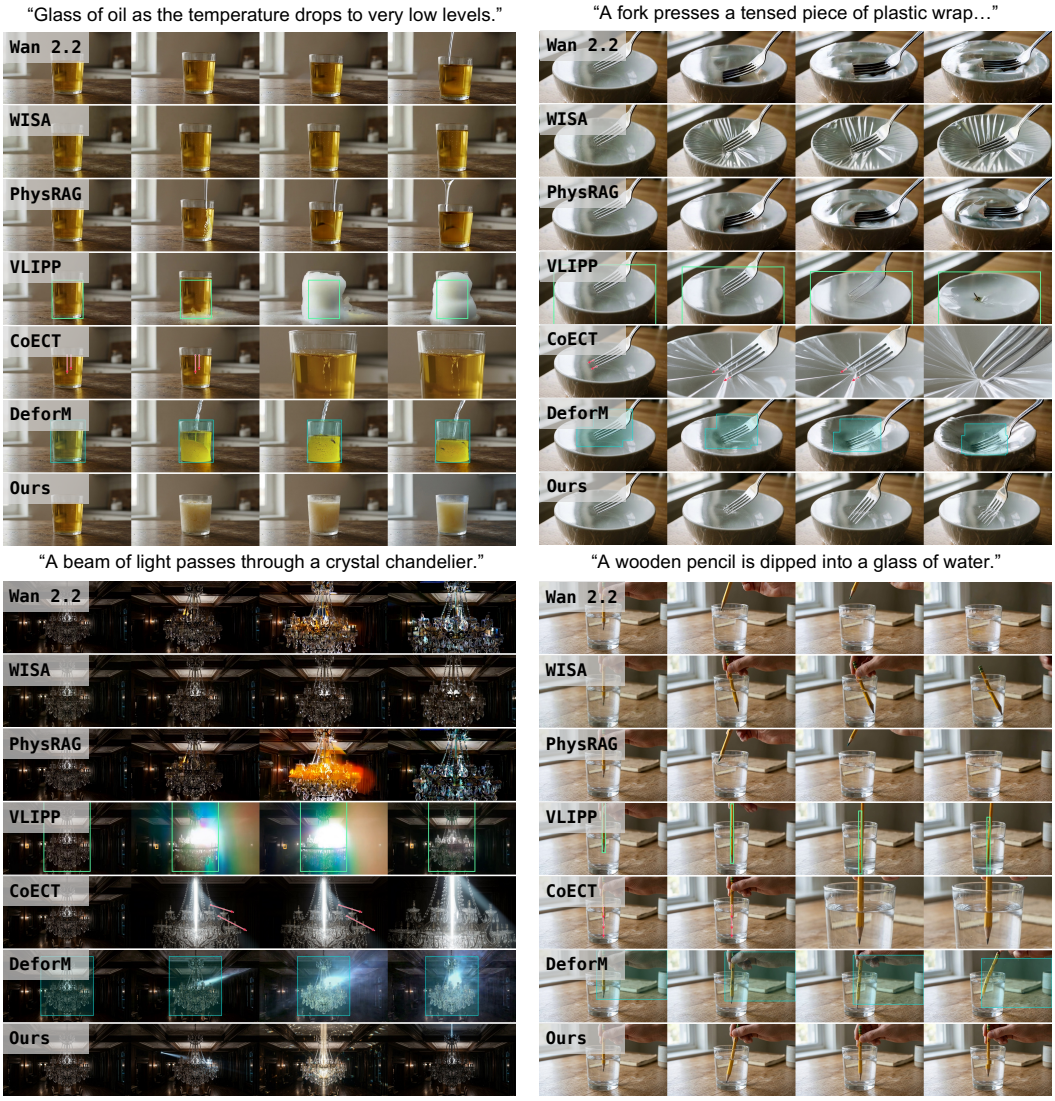

Figure 13: Qualitative comparison of generated videos across physics-aware methods and the video backbone. VLIPP, CoECT and DeforM exploits VLMs to generate explicit control signals (bounding box trajectory, drag signals and spatial-temporal masks, respectively). In contrast, PhysChain directly simulates visual states and refines using constraints, offering diverse physical domains and high fidelity.

A.3 Generalization Across Backbones

PhysChain separates VLM-based physical reasoning from image-based visual state prediction, so each component can use a different backbone. We test a 2×2 combination of image editing backbones (Gemini 3 Pro Image (Google DeepMind, 2025a) and Qwen-Image-2.1 (Qwen Team, 2026b)) and VLM reasoning backbones (Gemini 3.1 Pro and Qwen3-VL-8B (Bai et al., 2025)), with all other settings fixed. As shown in Tab. 2 and Fig. 14, all four configurations reproduce the intended dynamics, with the PhyGenBench score dropping by at most 0.026 when both backbones are open-source. This indicates that PhysChain does not rely on a specific backbone. Notably, the much smaller Qwen3-VL-8B performs comparably to Gemini 3.1 Pro, suggesting that the atomic constraints from the observable-cue memory keep verification tractable.

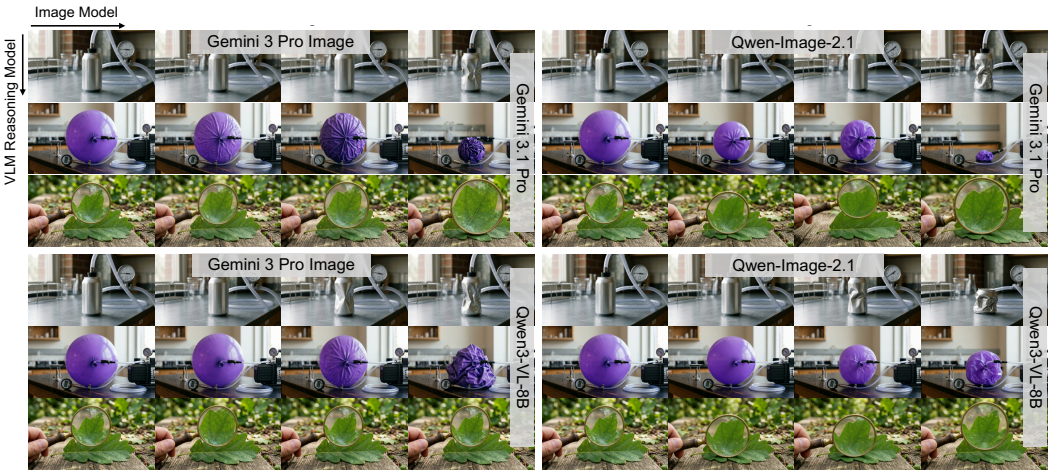

Figure 14: **Qualitative results across backbones.** Simulated visual states of PhysChain under the four combinations of VLM reasoning (rows: Gemini 3.1 Pro and Qwen3-VL-8B) and image editing (columns: Gemini 3 Pro Image and Qwen-Image-2.1) backbones, on can deformation, balloon deflation, and magnification through a lens. All configurations reproduce the intended physical phenomena, while differing mainly in appearance preservation and the smoothness of the progression across states.

Table 2: **Backbone generalization on PhyGenBench.** Configurations differ only in their image editing and VLM reasoning backbones. Values are PhyGenBench Avg. (↑).

VLM reasoning \ Image editing	Gemini 3 Pro Image	Qwen-Image-2.1
Gemini 3.1 Pro	0.754	0.741
Qwen 3-VL-8B	0.733	0.728

A.4 Additional Quantitative Results

For completeness, Tab. 3 reports the full numerical results across the five benchmarks visualized in Sec. 4, and Tabs. 4–8 place them alongside the results of video foundation models reported in each benchmark or its official leaderboard. Note that these reported results are obtained under the settings of the original evaluations (e.g., prompts and model versions at the time of evaluation), so the comparison is for reference rather than a strictly controlled one. Across the five benchmarks, PhysChain ranks first on PhyGenBench, PISA, and Morpheus, and second on Physics-IQ Verified, while remaining competitive on WorldModelBench, despite being built on a 5B video backbone.

Table 3: **Full quantitative results.** For completeness, we report all numerical results across the five benchmarks. Unavailable results are marked with "-". The best and second-best results are shown in bold and underlined, respectively.

Method	PhyGenBench					WorldModelBench			
	Mech. ↑	Optics ↑	Thermal ↑	Material ↑	Avg. ↑	Instr. ↑	Common. ↑	Physics ↑	Sum. ↑
Wan 2.2	0.579	0.610	0.477	0.490	0.547	1.98	1.43	4.49	7.90
WISA	0.525	**0.760**	0.578	0.533	0.610	2.03	1.35	4.36	7.74
PhysRAG	0.558	0.653	0.422	0.375	0.516	1.86	1.40	4.59	7.85
VLIPP	0.562	0.699	0.613	0.522	0.604	1.65	1.14	4.34	7.13
CoECT	0.656	0.714	0.653	0.597	0.659	1.16	0.46	4.77	6.39
DeforM	0.583	0.693	0.644	0.467	0.600	1.82	1.02	4.29	7.13
Ours-closed	**0.698**	0.758	**0.809**	**0.764**	**0.754**	**2.51**	1.82	4.71	**9.03**
Ours-open	0.696	0.731	0.759	0.734	0.728	2.24	**1.82**	**4.84**	8.91

Method	Physics-IQ						PISA			Morpheus		
	Mech. ↑	Fluid ↑	Optics ↑	Thermo ↑	Magnetism ↑	Full ↑	L2 ↓	CD ↓	IoU ↑	Dynamical ↑	Physical Inv. ↑	Avg. ↑
Wan 2.2	29.84	31.56	10.39	22.50	30.42	27.56	0.132	0.348	0.069	0.5023	0.3912	0.4468
WISA	36.21	35.88	23.19	46.29	27.25	34.74	0.149	0.434	0.059	0.5290	0.5765	0.5528
PhysRAG	35.93	24.31	14.68	29.88	14.48	29.79	0.164	0.462	0.065	0.6657	0.5476	0.6067
VLIPP	41.90	33.20	18.23	12.65	9.90	34.75	0.130	0.339	0.065	**0.7218**	0.5774	0.6496
CoECT	34.32	34.19	**25.90**	35.11	30.70	33.20	0.189	0.549	0.064	0.4434	0.6981	0.5708
DeforM	32.36	24.04	14.71	23.43	13.13	27.34	0.154	0.382	0.031	0.5079	0.5074	0.5077
Ours-closed	41.74	35.60	22.73	**56.23**	**34.13**	38.47	**0.120**	**0.300**	**0.073**	0.6829	0.6488	0.6659
Ours-open	**44.37**	**41.44**	24.31	38.11	29.34	**40.53**	0.137	0.340	0.051	0.7029	**0.7178**	**0.7104**

PhyGenBench. Tab. 4 additionally lists proprietary and open-source video foundation models and two preference-optimization methods, PhysHPO and PhyGDPO, whose results are taken from their papers. PHYSCHAIN achieves the highest average score of 0.75 among all entries, and improves the thermal and material categories most, which are the domains least covered by designated intermediate representations.

WorldModelBench. Tab. 5 compares with proprietary and open-source video foundation models. PHYSCHAIN closes most of the gap between its Wan2.2-5B backbone (7.90) and the proprietary models (8.24–9.27), surpassing MiniMax, Runway, and Luma with a total of 9.03. On the physics score, Ours-open reaches 4.84, the highest among all entries, while the remaining gap to Wan2.6 and Veo3 lies mainly in common sense, which depends on the generative prior of the video model rather than on the simulated physics.

Physics-IQ Verified. Tab. 6 compares with the official I2V leaderboard, for which only the full scores are reported. Starting from Wan2.2-5B (27.56), PHYSCHAIN reaches 40.53 with the open backbones, surpassing the larger Wan2.2-14B (32.2), MiniMax H3 (39.8), and Cosmos3-Nano (37.3), and second only to Cosmos3-Super (42.7). Notably, Ours-open exceeds Ours-closed here and on Morpheus, indicating that the gains stem from the constraint-based simulation rather than from the capacity of the closed backbones.

PISA. Tab. 7 compares with the video foundation models reported in the original benchmark. PHYSCHAIN achieves the lowest L2 and Chamfer distances among all entries, and the second-highest IoU after CogVideoX-5B-I2V, reflecting that the simulated states are verified against the free-fall progression before rendering.

Morpheus. Tab. 8 compares with the single-frame conditioning results reported in Morpheus, where only the combined score is available for the reported models. PHYSCHAIN achieves the highest combined score of 0.71 among all entries, exceeding Kling-2.5-Turbo (0.60) and COSMOS-predict2 (0.50), with both the dynamical and physical invariance components improved over its backbone.

Table 4: **PhyGenBench results.** Unmarked results for video foundation models are taken from PhyGenBench (Meng et al., 2025), while PhysHPO and PhyGDPO results are taken from their respective papers. Base model size refers to the video generation backbone only, excluding auxiliary models and method-specific modules; "-" denotes unavailable or unconfirmed sizes. The best and second-best results are shown in bold and underlined, respectively.

Method	Base model size	Mech.	Opt.	Therm.	Mat.	Avg.
Proprietary Video Foundation Model						
Kling (Kuaishou, 2025)	-	0.45	0.58	0.50	0.40	0.49
Pika (Pika Labs, 2024)	-	0.35	0.56	0.43	0.39	0.44
Gen-3 (Runway ML, 2024)	-	0.45	0.57	0.49	0.51	0.51
Open-source Video Foundation Model						
Lavie (Wang et al., 2025b)	-	0.30	0.44	0.38	0.32	0.36
Open-Sora v1.2 (Zheng et al., 2024)	1.1B	0.43	0.50	0.44	0.37	0.44
Vchitect 2.0 (Fan et al., 2025)	2B	0.41	0.56	0.44	0.37	0.45
CogVideoX-5B (Yang et al., 2025b)	5B	0.39	0.55	0.40	0.42	0.45
Wan 2.2† (Wan Team, 2025)	5B	0.58	0.61	0.48	0.49	0.55
Physics-aware Video Generation Model						
PhysHPO (Chen et al., 2025b)	5B	0.55	0.68	0.50	0.65	0.61
PhyGDPO (Cai et al., 2026)	14B	0.55	0.60	0.58	0.47	0.55
WISA† (Wang et al., 2025a)	14B	0.53	**0.76**	0.58	0.53	0.61
PhysRAG† (Cheng et al., 2026)	5B	0.56	0.65	0.42	0.38	0.52
VLIPP†,‡ (Yang et al., 2025a)	5B	0.56	0.70	0.61	0.52	0.60
CoECT†,‡ (Wang et al., 2026c)	5B	0.66	0.71	0.65	0.60	0.66
DeforM†,‡ (Li et al., 2026c)	5B	0.58	0.69	0.64	0.47	0.60
Ours-closed†	5B	**0.70**	<u>0.76</u>	**0.81**	**0.76**	**0.75**
Ours-open†	5B	<u>0.70</u>	0.73	<u>0.76</u>	<u>0.73</u>	<u>0.73</u>

† Results are obtained by us using the official PhyGenBench evaluation protocol, with the marked methods evaluated in the I2V setting using the same input images as ours for a fair comparison.
‡ For VLIPP, CoECT, and DeforM, we replace their VLM backbones with the same closed-source VLM as Ours-closed (Gemini 3.1 Pro) for a fair comparison.

Table 5: **WorldModelBench results.** Unmarked results are taken from WorldModelBench (Li et al., 2025b), except for Veo3 and Wan2.6, whose results are taken from Qwen-RobotWorld (Qwen Team, 2026a). Base model size follows Tab. 4. The best and second-best results are shown in bold and underlined, respectively.

Method	Base model size	WorldModelBench			
		Instr. ↑	Common. ↑	Physics ↑	Sum. ↑
Proprietary Video Foundation Model					
Wan2.6 (Alibaba Cloud, 2025)	-	2.50	<u>1.94</u>	<u>4.83</u>	**9.27**
Veo3 (Google DeepMind, 2025b)	-	**2.52**	1.93	4.80	<u>9.25</u>
KLING (Kuaishou, 2025)	-	2.32	**1.96**	4.82	9.10
Minimax (Li et al., 2025b)	-	2.28	1.92	4.72	8.92
Runway (Runway ML, 2024)	-	2.17	1.86	4.60	8.64
Luma (Luma AI, 2024)	-	1.98	1.76	4.49	8.24
Open-source Video Foundation Model					
CogVideoX-I2V (Yang et al., 2025b)	5B	1.78	1.12	4.17	7.08
OpenSora-I2V (Zheng et al., 2024)	1.1B	1.42	0.55	3.85	5.82
Wan 2.2† (Wan Team, 2025)	5B	1.98	1.43	4.49	7.90
Physics-aware Video Generation Model					
WISA† (Wang et al., 2025a)	14B	2.03	1.35	4.36	7.74
PhysRAG† (Cheng et al., 2026)	5B	1.86	1.40	4.59	7.85
VLIPP†,‡ (Yang et al., 2025a)	5B	1.65	1.14	4.34	7.13
CoECT†,‡ (Wang et al., 2026c)	5B	1.16	0.46	4.77	6.39
DeforM†,‡ (Li et al., 2026c)	5B	1.82	1.02	4.29	7.13
Ours-closed†	5B	<u>2.51</u>	1.82	4.71	9.03
Ours-open†	5B	2.24	1.82	**4.84**	8.91

† Results are obtained by us using the official WorldModelBench evaluation protocol.
‡ For VLIPP, CoECT, and DeforM, we replace their VLM backbones with the same closed-source VLM as Ours-closed (Gemini 3.1 Pro) for a fair comparison.

Table 6: **Physics-IQ Verified results.** Unmarked Full scores are taken from the official Physics-IQ Verified I2V leaderboard (Rädsch et al., 2026; Physics-IQ Verified, 2026). Category-level scores are not reported for these leaderboard entries and are therefore shown as "-". Base model size follows Tab. 4. The best and second-best results are shown in bold and underlined, respectively.

Method	Base model size	Physics-IQ Verified					
		Mech. ↑	Fluid ↑	Optics ↑	Thermo ↑	Magnetism ↑	Full ↑
Proprietary Video Foundation Model							
MiniMax H3 Max (MiniMax, 2026; fal.ai, 2026)	-	-	-	-	-	-	36.2
Grok Imagine Video (xAI, 2026)	-	-	-	-	-	-	34.8
Sora 2 (OpenAI, 2025)	-	-	-	-	-	-	26.5
P-Video (Rachwan et al., 2026)	-	-	-	-	-	-	25.3
Open-source Video Foundation Model							
Cosmos3-Super-Image2Video (NVIDIA, 2026)	64B*	-	-	-	-	-	**42.7**
MiniMax H3 (FL2VA) (MiniMax, 2026)	33B	-	-	-	-	-	39.8
Cosmos3-Nano (NVIDIA, 2026)	16B*	-	-	-	-	-	37.3
Hunyuan Video 1.5 (Tencent Hunyuan Foundation Model Team, 2025)	8.3B	-	-	-	-	-	33.4
Cosmos3-Edge (NVIDIA, 2026)	4B*	-	-	-	-	-	32.7
Wan 2.2-14B (Wan Team, 2025)	27B / 14B	-	-	-	-	-	32.2
CogVideoX-5B (Yang et al., 2025b)	5B	-	-	-	-	-	31.8
Kandinsky-WM 1.0 (Kandinsky Lab, 2026)	2B	-	-	-	-	-	30.8
Wan 2.2-5B† (Wan Team, 2025)	5B	29.84	31.56	10.39	22.50	30.42	27.56
Physics-aware Video Generation Model							
WISA† (Wang et al., 2025a)	14B	36.21	<u>35.88</u>	23.19	<u>46.29</u>	27.25	34.74
PhysRAG† (Cheng et al., 2026)	5B	35.93	24.31	14.68	29.88	14.48	29.79
VLIPP†,‡ (Yang et al., 2025a)	5B	<u>41.90</u>	33.20	18.23	12.65	9.90	34.75
CoECT†,‡ (Wang et al., 2026c)	5B	34.32	34.19	**25.90**	35.11	<u>30.70</u>	33.20
DeforM†,‡ (Li et al., 2026c)	5B	32.36	24.04	14.71	23.43	13.13	27.34
Ours-closed†	5B	41.74	35.60	22.73	**56.23**	**34.13**	38.47
Ours-open†	5B	**44.37**	**41.44**	<u>24.31</u>	38.11	29.34	<u>40.53</u>

† Results are obtained by us using the Physics-IQ Verified evaluation pipeline, with the marked methods evaluated in the I2V setting from the same benchmark switch-frame inputs as ours for a fair comparison.

‡ For VLIPP, CoECT, and DeforM, we replace their VLM backbones with the same closed-source VLM as Ours-closed (Gemini 3.1 Pro) for a fair comparison.

* Cosmos3 sizes are the officially reported total MoT parameter counts, including both the reasoner and generator towers, rather than generator-only counts. For Wan 2.2-14B, "27B / 14B" denotes total / active parameters per denoising step.

Table 7: **PISA results.** Unmarked evaluation scores are taken from the original PISA benchmark (Li et al., 2025a). Base model size follows Tab. 4. The best and second-best results are shown in bold and underlined, respectively.

Method	Base model size	PISA		
		L2 ↓	CD ↓	IoU ↑
Proprietary Video Foundation Model				
Sora (OpenAI, 2024b)	-	0.174	0.488	0.065
Kling-V1 (Kuaishou, 2025)	-	0.157	0.425	0.056
Kling-V1.5 (Kuaishou, 2025)	-	0.155	0.424	0.058
Runway Gen3 (Runway ML, 2024)	-	0.187	0.526	0.042
Open-source Video Foundation Model				
CogVideoX-5B-I2V (Yang et al., 2025b)	5B	0.138	0.366	**0.080**
DynamiCrafter (Xing et al., 2023)	1.4B	0.187	0.504	0.021
Pyramid-Flow (Jin et al., 2025)	-	0.175	0.485	0.062
Open-Sora (Zheng et al., 2024)	1.1B	0.175	0.502	0.069
Wan 2.2† (Wan Team, 2025)	5B	0.132	0.348	0.069
Physics-aware Video Generation Model				
WISA† (Wang et al., 2025a)	14B	0.149	0.434	0.059
PhysRAG† (Cheng et al., 2026)	5B	0.164	0.462	0.065
VLIPP†,‡ (Yang et al., 2025a)	5B	<u>0.130</u>	<u>0.339</u>	0.065
CoECT†,‡ (Wang et al., 2026c)	5B	0.189	0.549	0.064
DeforM†,‡ (Li et al., 2026c)	5B	0.154	0.382	0.031
Ours-closed†	5B	**0.120**	**0.300**	<u>0.073</u>
Ours-open†	5B	0.137	0.340	0.051

† Results are obtained by us using the PISA evaluation protocol, with the marked methods evaluated in the I2V setting using the same input images as ours for a fair comparison.

‡ For VLIPP, CoECT, and DeforM, we replace their VLM backbones with the same closed-source VLM as Ours-closed (Gemini 3.1 Pro) for a fair comparison.

Table 8: **Morpheus results.** Unmarked evaluation scores are the single-frame conditioning results reported in Morpheus (Tragoudaras et al., 2026). Avg. denotes the mean of the two component scores, $(\text{Dynamical} + \text{Physical Invariance})/2$; for unmarked baselines, only this average is reported, and unavailable component scores are denoted by "-". Base model size follows Tab. 4. The best and second-best results are shown in bold and underlined, respectively.

Method	Base model size	Morpheus		
		Dynamical ↑	Physical Invariance ↑	Avg. ↑
Proprietary Video Foundation Model				
Kling-2.5-Turbo (Kuaishou Technology, 2025)	-	-	-	0.60
Veo3-fast (Google DeepMind, 2025b)	-	-	-	0.49
Veo3 (Google DeepMind, 2025b)	-	-	-	0.46
Open-source Video Foundation Model				
COSMOS-predict2 (NVIDIA, 2025)	14B	-	-	0.50
WAN-2.1 (Wan Team et al., 2025)	14B	-	-	0.44
COSMOS-predict1 (NVIDIA et al., 2025)	14B	-	-	0.37
PyramidalFlow (Jin et al., 2025)	-	-	-	0.34
CogVideoX-5B (Yang et al., 2025b)	5B	-	-	0.32
LTX (HaCohen et al., 2024)	13B	-	-	0.28
Wan 2.2† (Wan Team, 2025)	5B	0.50	0.39	0.45
Physics-aware Video Generation Model				
WISA† (Wang et al., 2025a)	14B	0.53	0.58	0.55
PhysRAG† (Cheng et al., 2026)	5B	0.67	0.55	0.61
VLIPP†,‡ (Yang et al., 2025a)	5B	**0.72**	0.58	0.65
CoECT†,‡ (Wang et al., 2026c)	5B	0.44	<u>0.70</u>	0.57
DeforM†,‡ (Li et al., 2026c)	5B	0.51	0.51	0.51
Ours-closed†	5B	0.68	0.65	<u>0.67</u>
Ours-open†	5B	<u>0.70</u>	**0.72**	**0.71**

† Results are obtained by us using the Morpheus evaluation pipeline under the single-frame conditioning setting, with the marked methods using the same benchmark input images as ours for a fair comparison.

‡ For VLIPP, CoECT, and DeforM, we replace their VLM backbones with the same closed-source VLM as Ours-closed (Gemini 3.1 Pro) for a fair comparison.

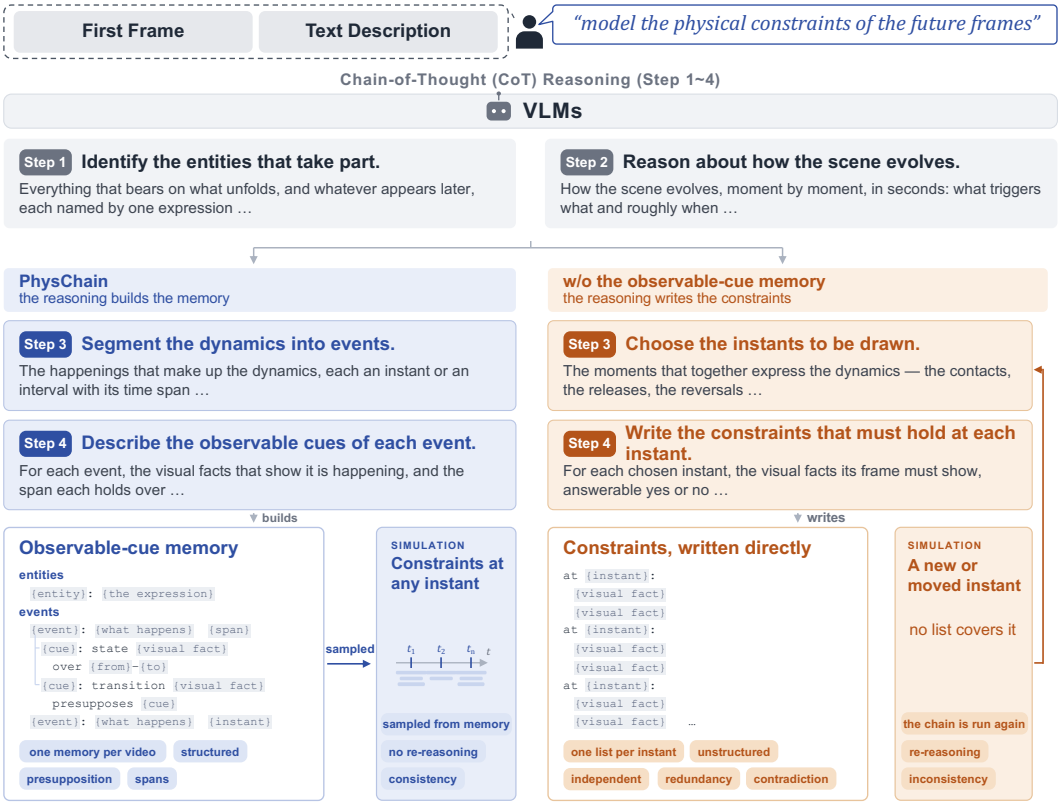

Figure 15: **Illustration of memory construction with chain-of-thought reasoning.** PHYSCHAIN reasons into the observable-cue memory, from which the constraints at any instant are sampled (left), while the same chain without the memory writes the constraints directly (right).

B ADDITIONAL IMPLEMENTATION DETAILS

B.1 MEMORY CONSTRUCTION WITH CHAIN-OF-THOUGHT (COT) REASONING

In Fig. 15, we illustrate the chain-of-thought (CoT) reasoning through which PHYSCHAIN constructs the observable-cue memory $\mathcal{M}$. Given the first frame S_0 and the text description y, VLMs reason through four chained steps: (1) identifying the entities $\mathcal{E}$ that take part, (2) reasoning about how the scene evolves over time, (3) segmenting the dynamics into events $\mathcal{V}$ with their time spans, and (4) describing the observable cues $\mathcal{O}$ of each event with the part of the event each holds over and the cue it presupposes. The chain is then transcribed into the memory $\mathcal{M} = (\mathcal{E}, \mathcal{V}, \mathcal{O})$ as a graph representation, from which the constraints at any instant are sampled by Eq. 1 throughout the simulation without further reasoning. In this representation, each event v has a time span $\sigma_v = [t^-, t^+] \subset \mathbb{R}$, which reduces to an instant when $t^- = t^+$. Each observable o associated with the event has a normalized span $\sigma_o \subseteq [0, 1]$ specifying the portion of σ_v it covers. A state observable describes a fact that holds over this span, whereas a transition observable describes a change that progresses over it.

Ablation on the observable-cue memory. In Tab. 1, we ablate the memory as follows: for a fair comparison, the setting without the memory keeps the CoT reasoning, reasoning through the same four steps but writing the constraints directly instead of building the memory, as illustrated on the right of Fig. 15. Steps 1 and 2 are identical, and Steps 3 and 4 are where the two settings differ. Step 3 chooses the instants to be drawn and reasons about what happens at each of them, rather than segmenting the dynamics into events $\mathcal{V}$ with their time spans; Step 4 writes the constraints that each of those instants must satisfy, rather than describing the observable cues $\mathcal{O}$ over the part of an event each holds. What this leaves is a flat list of constraints at each instant, unstructured across the video, with a fact that holds across several instants written anew in each. Over the simulation, an instant added or moved by the correction is covered by none of the lists and requires re-reasoning, which undermines the constraints as a consistent reference across the instants. The rest of the simulation is unchanged, apart from what the memory itself supplies to the verification and the revision.

B.2 SIMULATION TRACE

In Fig. 16, we trace the simulation of Sec. 3.3. We list ten of the 82 observables, each a single verifiable visual fact, with its kind, its event, and the observables it presupposes (top right), lay the eight events and the spans of all observables on the video time axis (left), and show the states of the three iterations with the constraints violated at each (right). [S] and [T] denote state and transition observables, γ_k the fraction of constraints satisfied at S_k, and the labels above each state mark the ones redrawn by Ψ and the instants moved, added, or removed by temporal correction.

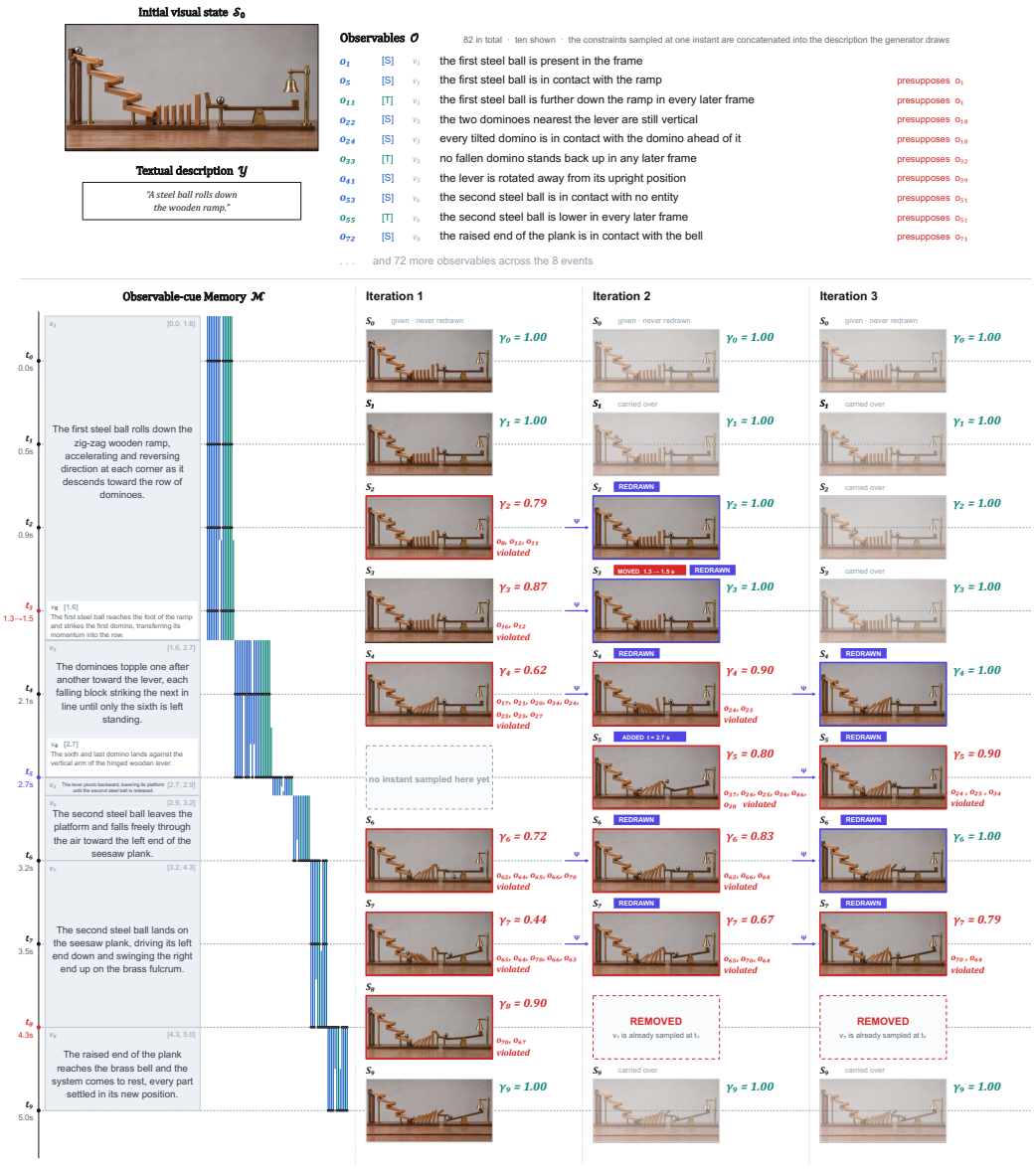

Figure 16: **Simulation trace.** From the initial visual state S_0 and text description y, the observable-cue memory $\mathcal{M}$ places eight events on the video timeline and associates each with state (blue) or transition (green) observables over temporal spans; 10 of 82 observables are shown (top, left). Sampling Π instantiates constraints from the sampled spans, and γ_k denotes the fraction satisfied at S_k. At each iteration, only states with violated constraints are redrawn, while temporal correction shifts, adds, or removes samples to better cover the event spans (right). Across three iterations, violations decrease from 25 to 14 to 5, while mean confidence increases from 0.82 to 0.91 to 0.97.

B.3 Confidence-adaptive Video Rendering

In Alg. 1, we outline the rendering algorithm of Sec. 3.4. We resample the states and their confidences onto the instants of the video model (lines 1–7), encode the resampled states and set the threshold of each latent frame (lines 9–14), and render the video by partially denoising the states from their thresholds (lines 16–28). E and D denote the encoder and the decoder of the VAE of the video model, and F a single denoising step conditioned on the first-frame latent $\mathbf{z}^0$ and the text description y.

Algorithm 1 Confidence-adaptive Video Rendering

Require: $S_0, y, \{(S_k, t_k, \gamma_k)\}_{k=1}^{n}, h, \tau_{\min}, \tau_{\max}$

1: $\{\mathbf{u}_k\}_{k=0}^{n-1} \leftarrow \mathrm{Corr}(\{S_k\}_{k=0}^{n})$
2: **for** $m = 1 \dots T$ **do** ▷ temporal resampling
3: $\quad \lambda_m \leftarrow (t_m - t_k)/(t_{k+1} - t_k)$
4: $\quad \bar{S}_m \leftarrow \mathcal{W}(S_k, \lambda_m \mathbf{u}_k)$
5: $\quad \hat{\gamma}_m \leftarrow (1 - \lambda_m)\gamma_k + \lambda_m \gamma_{k+1}$
6: $\quad \bar{\gamma}_m \leftarrow \hat{\gamma}_m \max_j \exp(-(t_m - t_j)^2/2h^2)$
7: **end for**
8:
9: $\mathbf{z}^0 \leftarrow E(S_0)$
10: $\bar{\mathbf{z}} \leftarrow E(\{\bar{S}_m\}_{m=1}^{T})$ ▷ L latent frames
11: $\boldsymbol{\epsilon} \sim \mathcal{N}(0, \mathbf{I})$
12: $\{\tau_{\mathrm{th}}^{\ell}\}_{\ell=1}^{L} \leftarrow \{\bar{\gamma}_{m_\ell} \cdot \tau_{\min} + (1 - \bar{\gamma}_{m_\ell}) \cdot \tau_{\max}\}_{\ell=1}^{L}$
13: $\tau_{\mathrm{start}} \leftarrow \max_\ell \tau_{\mathrm{th}}^{\ell}$
14: $\mathbf{z}_{\tau_{\mathrm{start}}} \leftarrow (1 - \tau_{\mathrm{start}}) \cdot \bar{\mathbf{z}} + \tau_{\mathrm{start}} \cdot \boldsymbol{\epsilon}$
15:
16: **for** $\tau = \tau_{\mathrm{start}} \dots 0$ **do** ▷ partial denoising
17: $\quad \bar{\mathbf{z}}_\tau \leftarrow (1 - \tau) \cdot \bar{\mathbf{z}} + \tau \cdot \boldsymbol{\epsilon}$
18: $\quad$ **for** $\ell = 1 \dots L$ **do**
19: $\quad\quad$ **if** $\tau > \tau_{\mathrm{th}}^{\ell}$ **then**
20: $\quad\quad\quad \mathbf{z}_\tau^{\ell} \leftarrow \bar{\mathbf{z}}_\tau^{\ell}$ ▷ overwrite to preserve the state
21: $\quad\quad$ **else**
22: $\quad\quad\quad$ keep $\mathbf{z}_\tau^{\ell}$ ▷ release to the prior
23: $\quad\quad$ **end if**
24: $\quad$ **end for**
25: $\quad \mathbf{z}_{\tau_{\mathrm{next}}} \leftarrow F(\mathbf{z}_\tau, \tau \mid \mathbf{z}^0, y)$ ▷ one denoising step to the next level
26: **end for**
27: $\mathbf{V} \leftarrow D(\mathbf{z}_0)$ ▷ decode the latents into frames
28: **Output:** Final video $\mathbf{V}$

B.4 Execution Time

In Tab. 9, we report the per-video execution time of PhysChain on PhyGenBench with the closed and open simulation backbones. With the closed backbones called through the Gemini API, memory construction takes $\sim$42 s and each predict-and-correct iteration $\sim$89 s, for a total simulation time of $\sim$219 s over 2.00 iterations on average. With the open backbones on a single NVIDIA A100 GPU, using 20 denoising steps for Qwen-Image-2.1, memory construction takes $\sim$29 s and each iteration $\sim$51 s, for a total of $\sim$128 s over 1.96 iterations on average. In both settings, the loop terminates early once all constraints are satisfied, and each state is predicted independently from S_0 rather than causally from the preceding state, which avoids error accumulation. Rendering is shared by both settings: the simulated states are converted into a 121-frame video (5 seconds at 24 fps, 832×480) on a single NVIDIA RTX 4090 GPU in $\sim$174 s. Since the confidence decays away from the simulated instants, rendering starts from $\tau_{\max}$ and runs 43 of the 50 denoising steps.

Table 9: Per-video execution time of PHYSCHAIN on PhyGenBench, with closed (Gemini 3.1 Pro + Gemini 3 Pro Image) and open (Qwen3-VL-8B + Qwen-Image-2.1) simulation backbones. The closed setting calls Gemini through its API; per-iteration figures are averaged over all executed iterations. Iterations stop early once all constraints are satisfied (2.00 and 1.96 iterations on average on closed and open model, respectively). The rendering stage is shared by both settings.

Stage	Component	Closed		Open	
		# Calls	Time (min)	# Calls	Time (min)
Simulation	Memory construction (once)	1	0.70	1	0.48
	Per iteration: prediction	3.48	1.48	4.3	0.60
	Per iteration: verification / revision	1.69		1.53	0.24
Simulation total ($\times$ avg. iterations)		11.34	3.65	12.4	2.13
Rendering	Wan2.2 TI2V-5B	–	2.91	–	2.91

C LIMITATIONS & FUTURE WORK

PHYSCHAIN enables tractable simulation of complex physical chains underlying entangled real-world dynamics, while several limitations remain. Although recent VLMs (Google DeepMind, 2026; Bai et al., 2025) exhibit strong reasoning capabilities, even reaching human-expert-level performance, our observations show that **(i)** ambiguity in physical intent occasionally yields a different yet plausible evolution from the reference, similar to variation in human interpretation. Despite reducing state prediction to the composition of observable facts, **(ii)** image synthesis leaves prediction errors. Confidence-adaptive rendering preserves reliable states while refining low-confidence ones with the video prior, but **(iii)** errors remain when the prior fails to recover the intended dynamics. Furthermore, our warp-based state resampling resolves artifacts from direct interpolation, while **(iv)** dense correspondences remain unreliable under occlusion, object disappearance, or large appearance changes. Finally, although relatively efficient on a consumer GPU as shown in Tab. 9, PHYSCHAIN is **(v)** not yet real-time. Extending physical chains to even longer horizons while maintaining reliable dynamics with real-time performance remains an important direction for future work.