PHYSICS BEFORE THE FIRST STEP: RECTIFYING THE TEXT EMBEDDING OF FROZEN VIDEO GENERATORS

Anonymous authors
Paper under double-blind review

ABSTRACT

Video generators trained at scale acquire a prior over natural motion, yet they still fail on actions whose outcome is dictated by physics. We observe that as flow matching and DiT backbones replace earlier designs, the structure of the generated video is decided earlier and earlier in the denoising process. This undermines methods that correct the latent during sampling, which need a window in which an error is visible and still open to repair. We turn instead to the text embedding, the one input read clean at every layer and every step, and find two properties that make it a sufficient and learnable target: an embedding fitted to a single real clip reproduces that clip under every initial noise, and the displacement to it is spread across the whole embedding, so a prediction that lands near it suffices. Building on these, we propose PhysRect: for physical events cut from real video, it fits the embedding under which a frozen generator reproduces the clip, keeping the fits of different mechanisms apart with a separability term. The fits are then distilled into a VLM-gated mixture of mechanism experts that corrects the embedding once before sampling. On VideoPhy-2, PhysRect reduces the physics error of Wan2.2 by 51%, matches a 14B model fine-tuned on physical video at one fourteenth of its generation time with no inference overhead.

1 INTRODUCTION

Physically plausible video generation has undergone a paradigm shift in recent years, moving from physics placed outside the model to physics learned implicitly from data. Traditional approaches either simulate the dynamics with a physics engine and render the result, or learn a dynamics predictor on synthetic toy domains (Battaglia et al., 2016; Watters et al., 2017; Le Guen & Thome, 2020; Wu et al., 2016; Yi et al., 2020), and neither transfers beyond the phenomena it was built for. In contrast, video foundation models scaled to billions of parameters and massive video data (Yang et al., 2025b; Kong et al., 2024; HaCohen et al., 2024; Wan et al., 2025) learn the regularities of natural motion, and dynamics such as falling, sliding and flowing emerge without any explicit physical model. However, recent benchmarks show that they still fail frequently on actions with a physically determined outcome (Bansal et al., 2025; Meng et al., 2024; Motamed et al., 2025).

Recent methods differ in where the correction enters the generator. One line changes the weights, fine-tuning on annotated or simulated physical video or aligning the model to physics-based rewards (Wang et al., 2026a; Li et al., 2025; Zhang et al., 2025b; Cai et al., 2025). Yet the pretraining video already obeys physics and scaling it has not taught physical laws (Kang et al., 2025; Motamed et al., 2025), and reward alignment is prone to reward hacking (Skalse et al., 2022; Clark et al., 2024; Liu et al., 2026a). The other line keeps the weights and intervenes at inference, on the initial noise (Foo et al., 2026), on the latent at each denoising step (Zhang et al., 2025a; Yang et al., 2025a; Chefer et al., 2025; Jang et al., 2026), or on the prompt (Xue et al., 2025). A correction on the latent, however, requires a window in which the physics is already visible and yet still open to change.

Decoding the clean estimate along the sampling path shows that the layout is settled within the first fifth of the run in diffusion transformers, against a third to a half in earlier UNet generators (Fig. 1a, bottom) (Balaji et al., 2022; Choi et al., 2022). In Wan2.2 the layout is visible after one step and settled by the third of its fifty (Fig. 1a, top). If a correction on the latent is confined to these few steps, the text embedding offers an alternative. The generator reads it clean at every layer and every step, so a correction fixed in it before sampling acts throughout the run and adds nothing to its

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.

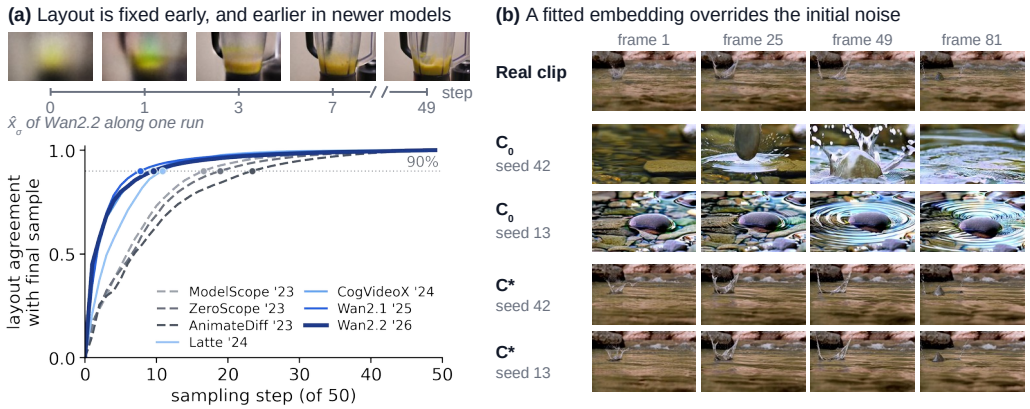


Figure 1: **(a)** Top: the clean estimate $\hat{x}_\sigma$ decoded along one Wan2.2 run, with the layout already visible at step 1. Bottom: low-frequency agreement between $\hat{x}_\sigma$ and the final sample for seven generators on the same 100 prompts, with dots marking 90% agreement. **(b)** Two seeds give two different scenes under the prompt embedding C_0 and the same reference clip under the fitted embedding C^*.

cost. We further find that the text embedding has two properties that make a correction there both sufficient and learnable. **Dominance**: an embedding fitted to a single real clip reproduces that clip under every initial noise (Fig. 1b). **Redundancy**: the displacement from the prompt embedding to the fitted one is spread across the whole embedding and exhibits high-rank characteristics in every row, which makes the correction robust to approximation and hence learnable.

Building on these, we propose PhysRect, which corrects the physics of a frozen video generator by rectifying its text embedding. The method learns from physical events: a real clip cut to the extent of one physical action, from onset to outcome, with a mask over the moving subject and a complete description. We build PHYSEVENTS, a pipeline that extracts such events from real video and yields 32,828 events over 483 actions. For each event we fit the embedding under which the generator reproduces the clip, with a separability term so that events of different mechanisms, such as rigid motion and fluid, displace the embedding in different directions. We then distill the fits into a VLM-gated mixture of experts that predicts the displacement from the prompt embedding and the mechanisms the prompt carries. At inference the text embedding is corrected once before sampling with the generator itself unchanged. On VideoPhy-2, PhysRect lowers the mean physics error of Wan2.2 from 1.43 to 0.71 and raises quality from 3.18 to 3.99, with the fraction of videos free of any physics error rising from 26% to 55%. Our contributions are as follows.

- We show that the window in which physics can be corrected on the latent is closing in modern generators, and identify the text embedding as the alternative, with two properties, dominance and redundancy, that make a correction there sufficient and learnable.
- We build PHYSEVENTS, a pipeline that isolates 32,828 physical events from real video, each with a subject mask and a description, with every stage validated against human labels.
- We propose PhysRect, which fits an embedding per physical event with a separability term, distills the fits into a VLM-gated mixture of experts, and applies the correction once before sampling, so a frozen generator gains physical consistency at no cost per step.

2 RELATED WORK

Physical plausibility in video generation. Video generators trained at scale get appearance right and motion wrong, and scaling does not close the gap: the models generalize case by case from nearby training clips rather than from a law (Kang et al., 2025). Most works retrain the generator. WISA (Wang et al., 2026a) fine-tunes on 32K clips labeled with 17 phenomena and routes attention heads by phenomenon, PISA (Li et al., 2025) post-trains on falling objects, others fine-tune on simulator-rendered footage (Zhao et al., 2025), VideoREPA (Zhang et al., 2025b) aligns intermediate features to a video foundation model, and a growing line optimizes preferences under a physics reward (Wang et al., 2025; Cai et al., 2025; Meng et al., 2026). All of these cost a training run

per backbone and inherit whatever the reward or annotation calls physics. Simulator-based methods avoid the reward by computing the answer: RealWonder (Liu et al., 2026b) and PSIVG (Foo et al., 2026) reconstruct the scene, simulate it, and pass the simulated motion to the generator as optical flow through the initial noise. Physics is exact where the simulator applies, but the conditioning is spatial, and a text prompt supplies no scene to reconstruct. The remaining option is to steer the denoising trajectory: VideoJAM (Chefer et al., 2025) guides each step with the model's own optical-flow prediction, MotionPrompt (Nam et al., 2024) back-propagates a flow discriminator into the prompt tokens during sampling, and Self-Refining Sampling (Jang et al., 2026) re-noises and re-denoises the latent at fixed noise levels with the model as its own critic. Each pays extra passes per step, and each starts on a latent whose motion is already largely fixed.

Acting on the text input. The prompt is the one input the generator reads clean at every step and every layer, and it can be changed in two ways. In words: PhyT2V (Xue et al., 2025) has a language model state the governing law and rewrite the prompt over several rounds of generate-and-caption, DiffPhy (Zhang et al., 2025a) and VLIPP (Yang et al., 2025a) plan the motion in text before generating, SDG (Hao et al., 2025) writes a prompt that violates the physics and guides away from it, and chain-of-events methods (Wang et al., 2026c;b) order the prompt causally. Or in the embedding itself: textual inversion fits token vectors to a few photographs (Gal et al., 2023a), and encoders replace the fit with a single forward pass (Gal et al., 2023b; Wei et al., 2023). For video, RichSpace (Cao et al., 2025) interpolates embeddings to reach prompts no text encoder produces, MotionPrompt (Nam et al., 2024) and DATE (Na et al., 2025) re-optimize the embedding during sampling at several times the sampling cost, and PSIVG (Foo et al., 2026) fits an offset on the foreground tokens per instance to bind texture to a simulated motion. Words can only say what a caption can say; the embedding methods fit per instance or per step, and target identity, style or prompt adherence. We fit the embedding against a real video clip of the event, distill the fit into one forward pass before sampling, and compose it from one sub-network per physical mechanism.

3 METHOD

3.1 PRELIMINARIES

Rectified flow on video latents. A video $x \in \mathbb{R}^{T\times H\times W\times 3}$ is encoded by a frozen VAE $\mathcal{E}$ into a latent $z = \mathcal{E}(x) \in \mathbb{R}^{t\times h\times w\times d_z}$ with $t = T/r_t$, $h = H/r_s$, $w = W/r_s$, and decoded by $\mathcal{D}$. Generation runs on z. Flow matching (Lipman et al., 2023) trains a network to predict the velocity of a path from noise to data. When the path is straight, the velocity is constant along it, a case known as rectified flow (Liu et al., 2023; Esser et al., 2024), which current video models adopt as their training objective (Wan et al., 2025; Kong et al., 2024). We write the noise level as $\sigma \in [0, 1]$ and the noisy latent as

$$z_\sigma = (1-\sigma)\, z + \sigma\, \epsilon, \qquad \epsilon \sim \mathcal{N}(0, I), \tag{1}$$

so that $\sigma = 1$ is pure noise and $\sigma = 0$ is data. A network v_θ is trained to predict the velocity of this path,

$$\mathcal{L}_{\mathrm{FM}} = \mathbb{E}_{\sigma,z,\epsilon}\left[\, \|v_\theta(z_\sigma, \sigma, C) - (\epsilon - z)\|^2 \right], \tag{2}$$

where C is the conditioning. Since the target $\epsilon - z$ does not depend on σ, the network is supervised by the same target at every noise level. Sampling integrates $dz_\sigma/d\sigma = v_\theta(z_\sigma, \sigma, C)$ from $\sigma = 1$ to 0 over S steps at levels $\sigma_1 = 1 > \cdots > \sigma_S > 0$. The clean latent implied by the state at σ is

$$\hat{z}_\sigma = z_\sigma - \sigma\, v_\theta(z_\sigma, \sigma, C), \tag{3}$$

and its decoding $\hat{x}_\sigma = \mathcal{D}(\hat{z}_\sigma)$ is the clean estimate in pixels. With classifier-free guidance (Ho & Salimans, 2022), the velocity used at each step is $v_\theta(z_\sigma, \sigma, \varnothing) + g\,[\, v_\theta(z_\sigma, \sigma, C) - v_\theta(z_\sigma, \sigma, \varnothing)\,]$ with guidance scale g.

Text embedding in the diffusion transformer. The velocity network v_θ is a diffusion transformer (Peebles & Xie, 2023): the latent is patchified into N tokens, and a prompt y is encoded once by a frozen text encoder τ into a text embedding

$$C = \tau(y) \in \mathbb{R}^{L\times d}, \tag{4}$$

the output of the text encoder, with one d-dimensional row per prompt token ($d = 4096$ for Wan). Every one of the B blocks applies self-attention over the latent tokens and then cross-attention in

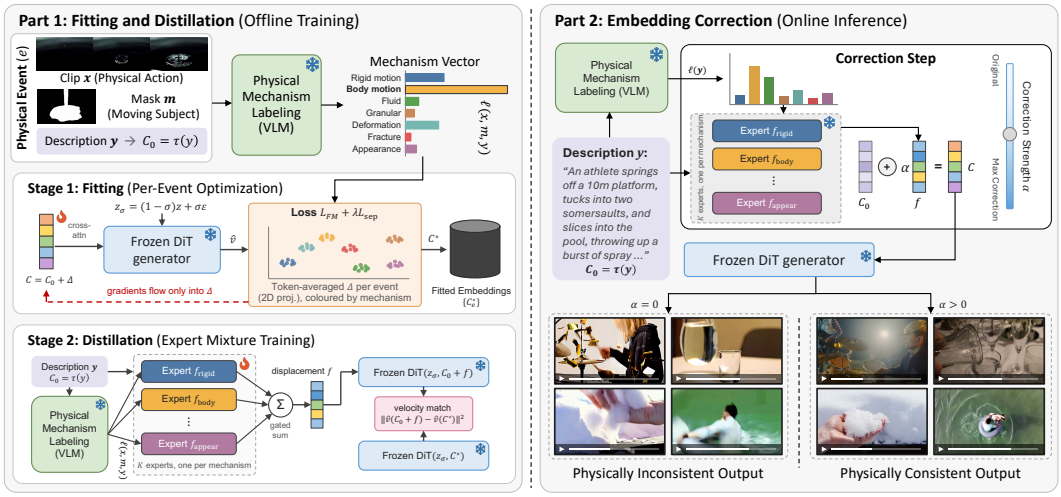


Figure 2: **Overview of PhysRect.** Left, training. Each physical event $e = (x, m, y)$ is labeled with its mechanism vector ℓ by a vision-language model. Stage 1 fits, for each event, the embedding $C^* = C_0 + \Delta$ under which the frozen generator reproduces the clip, with the separability term keeping the displacements of different mechanisms apart. Stage 2 distills the fits into a mixture of experts, one per mechanism, gated by ℓ and trained to match the velocity the generator produces under C^*. Right, inference. A prompt is encoded once and labeled once; the gated experts produce the correction, scaled by α, and the generator samples under the corrected embedding.

which the latent tokens query C through the block's own projections $W_k^{(b)}C$, $W_v^{(b)}C$. C depends on neither σ nor z_σ, and every block reads it, so the same noise-free matrix enters every layer at every step, and a change to C reaches all of them without any change to θ. The generator is the map $v_\theta(z_\sigma, \sigma, C)$ with θ frozen throughout this paper, and all we change is C.

Physical events. Let $V = (v_1, v_2, \dots)$ be a source video and let $\mathcal{P}(V) = \{[s_i, u_i]\}$ denote the frame intervals over which a single subject undergoes a single physical process from onset to outcome. A window $x = v_{a:a+T}$ of T frames is a physical event if it contains exactly one such interval, $|\{i : [s_i, u_i] \subseteq [a, a+T)\}| = 1$. An event is represented as a triple $e = (x, m, y)$, with m a mask over the subject in every frame of x and y a description of the subject and of the process through to its outcome. Each event carries a mechanism vector $\ell \in [0, 1]^K$ over K physical mechanisms, read from (x, m, y) by a vision-language model, whose entries grade the presence of each mechanism from absent to dominant.

3.2 PhysRect Overview

We propose PhysRect, which improves the physical consistency of a frozen video generator by acting on its text embedding alone. The overall design is shown in Figure 2. Given a prompt y with embedding $C_0 = \tau(y)$, the generator samples under a corrected embedding

$$C = C_0 + \alpha\, f\big(C_0, \ell(y)\big), \tag{5}$$

where f is a learned predictor, $\ell(y)$ is the mechanism vector of the prompt, and α sets the strength of the correction.

Because C enters every block at every step and carries no noise at any σ, a correction placed in it acts from the first step and at every layer at no cost per step, and since the text embedding is dominant enough to override the initial noise, the correction can be as large as the physics requires. f is trained in velocity space (Eq. 9): at every noise level it is asked to make the generator move as it does under the fitted embedding, and since that target is the same at every level, one correction computed before sampling holds along the whole path. Sec. 3.3 gives f and its training, and Sec. 3.4 the pipeline that supplies the physical events it is trained on.

3.3 EMBEDDING RECTIFICATION

Event-specific embedding fitting. For each event $e = (x, m, y)$, we look for the embedding under which the frozen generator reproduces its clip x. Starting from $C_0 = \tau(y)$, we hold θ fixed and optimize the embedding alone:

$$C^* = \arg\min_C \; \mathbb{E}_{\sigma,\epsilon}\big\|v_\theta(z_\sigma, \sigma, C) - (\epsilon - z)\big\|^2 + \lambda\, \mathcal{L}_{\text{sep}}(\bar{\Delta}), \qquad \Delta = C - C_0, \tag{6}$$

where $z = \mathcal{E}(x)$, $\bar{\Delta} = \frac{1}{L}\sum_{i=1}^{L} \Delta_i$ is the mean over the L rows of the displacement, and $\mathcal{L}_{\text{sep}}(\bar{\Delta}) = \|\bar{\Delta} - \mu_{c^*}\|^2 - \|\bar{\Delta} - \mu_{c'}\|^2$, with μ_c the running mean of $\bar{\Delta}$ over events whose dominant mechanism is c, $c^* = \arg\max_c \ell_c$ the dominant mechanism of the event and c' the nearest other mechanism. The first term is the flow matching loss on a single clip; the second pulls the displacement towards the prototype μ_{c^*} of its own mechanism and away from $\mu_{c'}$, so that events of different mechanisms move the embedding in different directions. Writing the first term as $\mathcal{L}_{\text{FM}}$, the weight is set once at the start of the fit, so that the separability term's gradient on the embedding is a fraction γ of the flow matching term's,

$$\lambda = \gamma\, \frac{\|\nabla_C \mathcal{L}_{\text{FM}}\|}{\|\nabla_C \mathcal{L}_{\text{sep}}\|}\Big|_{\text{step } 0}. \tag{7}$$

This differs from textual inversion in what is fitted and why: not a new token that names an appearance, but a displacement of the whole embedding that carries a motion, and one constrained to be separable by mechanism (Fig. 3) so that it can be learned from in the next step. Each event yields a fitted embedding C^* and its displacement $\Delta^* = C^* - C_0$.

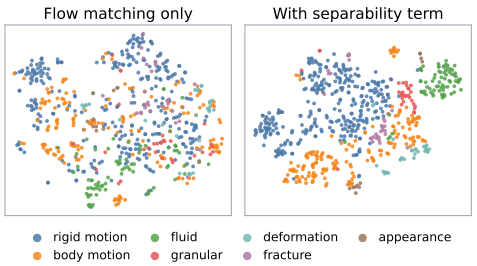


Figure 3: Fitted embeddings C^*, pooled over rows and embedded with t-SNE, colored by dominant mechanism. Under flow matching alone (left) the mechanisms are not separable and a linear probe on the displacement reaches 0.54; with the separability term (right) the same probe reaches 1.00.

Distilling the fits into mechanism experts. Fitting is per event and too slow for inference, so a predictor learns to produce the displacement from the prompt's embedding alone. We build f as a mixture of experts, one per mechanism. Each expert f_c is a small transformer over the rows of C_0 whose output layer is initialized to zero. The experts share no parameters because the correction for a mechanism is a rule of its own: a prompt that carries two mechanisms should receive both rules, and a single network trained on all of them would entangle them. The experts are combined by the mechanisms the prompt carries:

$$f(C_0, \ell) = \frac{\sum_c w_c\, f_c(C_0)}{\sum_c w_c}, \qquad w_c = \mathbf{1}[\ell_c > 0]. \tag{8}$$

so that a prompt with several mechanisms receives the average of their rules and one with a single mechanism receives that rule alone. The experts are trained not to regress Δ^* but to make the generator behave as it does under C^*:

$$\mathcal{L}_f = \mathbb{E}_{e,\sigma,\epsilon}\big\|v_\theta\big(z_\sigma, \sigma, C_0 + f(C_0, \ell)\big) - v_\theta(z_\sigma, \sigma, C^*)\big\|^2, \tag{9}$$

with the same z_σ on both sides. Matching velocities rather than Δ^* itself discards the components of Δ^* that do not change the velocity.

Inference. A prompt y is encoded into C_0 and labeled with its mechanisms $\ell(y)$ by a vision-language model. The experts are combined with the graded weights $w_c = \ell_c$, and the generator samples once from the embedding of Eq. 5 with nothing else changed. The mechanism vector is read from the prompt alone, $\ell(y)$, where an event is labeled from its clip, mask and description, $\ell(x, m, y)$. The two agree because the description is written from the clip and states the process through to its outcome (Sec. 3.4), so the mechanisms a prompt names are the ones its clip shows.

3.4 THE PHYSEVENTS PIPELINE

Motivation. The fit in Sec. 3.3 needs a real clip of a single physical event, with the subject located and a description that asks the generator for exactly that event. Data used for physics-aware generation comes in three kinds: (1) real video curated by phenomenon, e.g., WISA-32K (Wang et al.,

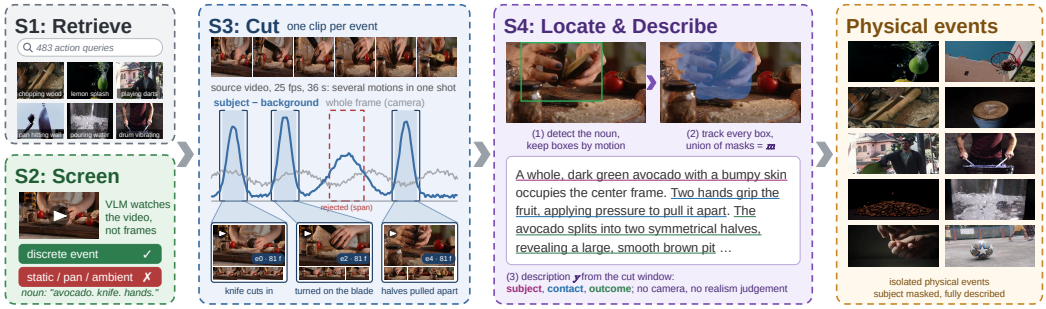


Figure 4: **Illustration of PHYSEVENTS.** Four steps take a source video to isolated physical events. The example shows why the cut is per event rather than per source: one 36-second shot contains three distinct motions, the motion signal peaks on each, and each peak becomes its own T-frame clip with a subject mask and a description. A fourth candidate is rejected by the span gate.

2026a) and PhyHQ (Zhang et al., 2025a); (2) post-training sets of one phenomenon, e.g., PISA (Li et al., 2025); (3) simulator-rendered video (Zhao et al., 2025).

Video curated by phenomenon is cut at shot boundaries, so a clip may hold several motions, and the subject's location is not recorded. Post-training sets cover a single phenomenon, and simulator-rendered video lacks the appearance of real footage. We therefore build PHYSEVENTS, which produces physical events as defined in Sec. 3.1.

Pipeline. As shown in Fig. 4, the pipeline has four steps. One rule holds throughout: every label is read from the pixels, never from the query that surfaced the footage, since a query specifies the intended content and not what the source contains.

Step 1: Retrieval. Stock footage is searched with 483 physical actions written by a language model and phrased like action recognition labels. Sources are kept only if they are landscape and at least the generation resolution, so that no clip is upscaled to the resolution the generator has to reproduce.

Step 2: Screening. A vision-language model watches each source as video and keeps it only if a discrete event with a start and an end is visible, returning a noun for what moves. Watching the video rather than sampled frames is what separates a throw from a slow reach and a moving subject from a moving camera. Its accuracy against human labels is 83%.

Step 3: Cutting. For each frame, the difference inside the detected subject is compared with the difference outside it, which gives a motion signal over time. Camera motion present in both is subtracted out, and the subject's own motion remains. Each event appears as a peak of this signal. One T-frame clip is cut per peak, so a source with several motions yields one clip per motion rather than being merged into one. Their accuracy against human labels is 82%.

Step 4: Localization and description. An open-vocabulary detector given the noun finds the subject. Boxes are kept by the motion they contain, tracked by a video segmenter, and unioned into m. Last, the vision-language model writes the description y from the cut clip, covering the subject, its contact and the outcome. The result is 32,828 events over 483 actions.

4 EXPERIMENT

4.1 EXPERIMENTAL SETUP

Setup. We evaluate on the 591 test prompts of VideoPhy-2 (Bansal et al., 2025), generating one video per prompt. Videos are scored with JudgeFit (Cao et al., 2026), which calibrates a vision-language judge against human physics annotations and discovers, for each judge, the set of error dimensions it can reliably assess. We use its Gemini 3 Flash (Google DeepMind, 2025) configuration, the judge with the highest agreement with human labels, which rates five physics error dimensions: object coherence, material properties, optics, kinematics and dynamics, and object interaction, each from 0 to 5 with lower being better, and an overall quality score from 1 to 5 with higher being better.

Table 1: **Physical consistency on VideoPhy-2.** JudgeFit scores over the 591 test prompts: five physics error dimensions (0 to 5, lower is better), their mean, and overall quality (1 to 5, higher is better). Time is the average generation time per video in seconds on the same device. Subscripts are standard errors of the mean over the 591 prompts. Best in bold.

Method	Backbone	Time (s)	Coh.	Mat.	Opt.	Kin.	Int.	Mean ↓	Quality ↑
Video generators									
CogVideoX-5B (Yang et al., 2025b)	–	116.5	$0.90_{\pm.05}$	$0.55_{\pm.04}$	$0.29_{\pm.03}$	$1.29_{\pm.06}$	$1.27_{\pm.06}$	$0.860_{\pm.042}$	$3.82_{\pm.05}$
Wan2.1-1.3B (Wan et al., 2025)	–	98.9	$1.46_{\pm.06}$	$1.01_{\pm.05}$	$0.53_{\pm.03}$	$1.74_{\pm.07}$	$1.74_{\pm.07}$	$1.295_{\pm.051}$	$3.37_{\pm.06}$
Wan2.1-14B (Wan et al., 2025)	–	479.5	$0.80_{\pm.05}$	$0.54_{\pm.04}$	$0.33_{\pm.03}$	$1.15_{\pm.06}$	$1.17_{\pm.06}$	$0.796_{\pm.042}$	$3.95_{\pm.05}$
Wan2.2-5B (Wan et al., 2025)	–	38.9	$1.52_{\pm.06}$	$1.03_{\pm.05}$	$0.74_{\pm.04}$	$1.96_{\pm.06}$	$1.91_{\pm.06}$	$1.431_{\pm.048}$	$3.18_{\pm.06}$
Fine-tuning and alignment									
VideoREPA (Zhang et al., 2025b)	CogVideoX-5B	116.0	$0.84_{\pm.05}$	$0.61_{\pm.04}$	$0.31_{\pm.03}$	$1.33_{\pm.06}$	$1.25_{\pm.06}$	$0.866_{\pm.042}$	$3.80_{\pm.06}$
PhyGDPO (Cai et al., 2025)	Wan2.1-1.3B	64.2	$0.99_{\pm.05}$	$0.77_{\pm.05}$	$0.51_{\pm.03}$	$1.40_{\pm.06}$	$1.37_{\pm.06}$	$1.009_{\pm.047}$	$3.75_{\pm.06}$
WISA (Wang et al., 2026a)	Wan2.1-14B	518.0	$0.73_{\pm.05}$	$0.55_{\pm.04}$	$0.34_{\pm.03}$	$1.15_{\pm.06}$	$1.09_{\pm.06}$	$0.774_{\pm.042}$	$3.97_{\pm.05}$
Inference-time intervention									
PhyT2V (Xue et al., 2025)	Wan2.2-5B	146.8	$1.74_{\pm.06}$	$1.24_{\pm.05}$	$0.90_{\pm.04}$	$2.09_{\pm.06}$	$2.08_{\pm.06}$	$1.610_{\pm.048}$	$3.05_{\pm.05}$
Self-Refining (Jang et al., 2026)	Wan2.2-5B	44.4	$2.76_{\pm.05}$	$2.02_{\pm.05}$	$2.14_{\pm.05}$	$2.81_{\pm.05}$	$2.92_{\pm.05}$	$2.530_{\pm.043}$	$2.13_{\pm.05}$
PhysRect (ours)	Wan2.2-5B	38.3	$\mathbf{0.67}_{\pm.04}$	$\mathbf{0.49}_{\pm.04}$	$\mathbf{0.21}_{\pm.02}$	$\mathbf{1.10}_{\pm.06}$	$\mathbf{1.06}_{\pm.06}$	$\mathbf{0.705}_{\pm.040}$	$\mathbf{3.99}_{\pm.06}$

We report the five dimensions, their unweighted mean, and quality. The judge was not used at any stage of training.

Baselines. We compare against four video generators, CogVideoX-5B (Yang et al., 2025b), Wan2.1-1.3B, Wan2.1-14B and Wan2.2-5B (Wan et al., 2025), three methods that fine-tune or align the generator, VideoREPA (Zhang et al., 2025b), PhyGDPO (Cai et al., 2025) and WISA (Wang et al., 2026a), and two that intervene at inference, PhyT2V (Xue et al., 2025) and Self-Refining (Jang et al., 2026). Each method is run with its released weights and code on its own backbone.

Implementation Details. We use Wan2.2-TI2V-5B (Wan et al., 2025) as the frozen generator, at 832×480 with 81 frames, 50 UniPC steps and guidance scale 5. The text encoder is umT5-XXL with $d = 4096$. Embeddings are fitted in two stages: a per-event fit with Adam (Kingma & Ba, 2015) at learning rate 10^{-3} for 250 steps, each accumulating 8 draws of (σ, ϵ), followed by a joint refit of all events from this warm start with the separability term, whose weight is set by Eq. 7 with $\gamma = 0.1$. The mixture has one expert per mechanism, each a two-layer transformer of width 512 with 8 heads over the rows of C_0, 73.5M parameters in total. The experts are trained on 576 fitted events with AdamW (Loshchilov et al., 2017) at learning rate 3×10^{-5} for 2,000 steps, about 6 GPU-hours. At inference we use $\alpha = 0.5$. Further details are in Appendix B.

4.2 Main results

Quantitative analysis. Table 1 compares PhysRect with four video generators and five physics-aware methods on the 591 VideoPhy-2 test prompts. PhysRect reduces the mean physics error of Wan2.2 from 1.43 to 0.71 and raises quality from 3.18 to 3.99, and the reduction holds on every dimension, from 44% on kinematics to 72% on optics. The fraction of videos with no error on any dimension rises from 26% to 55%, and the fraction with a severe error (3 or above) on any dimension falls from 52% to 28%. Among the fine-tuning methods, PhyGDPO reduces the error of its backbone by 0.29, and VideoREPA and WISA score level with theirs. PhysRect reduces it by 0.73 with the generator frozen and reaches the lowest error and highest quality in the table, matching WISA and Wan2.1-14B at a 5B backbone and one fourteenth of the generation time. The inference-time methods on the same backbone do not improve on it: PhyT2V adds 0.18 to the error after three rounds of prompt refinement at 3.8 times the generation time. PhysRect adds no time per video, 38.9 s against 38.3 s, where the other methods range from 44 s to 518 s.

Qualitative analysis. Figure 5 compares the six methods on three prompts at the same four frames. Overall, PhysRect tends to keep objects intact and to play the described process through to its outcome, while the other methods more often depart from it, with objects passing through one another, a subject separating from the object it holds, or a scene that stays still where the prompt describes motion. For example, in the fire scene (bottom), the water stream of Wan2.2 and VideoREPA passes through the flame without affecting it, PhyGDPO shows no water at all, and under WISA the flame grows larger under the spray, whereas PhysRect brings it down. In the breaststroke scene (middle), the arms of Wan2.2 and PhyGDPO flail without a stroke pattern, and PhysRect carries the arms

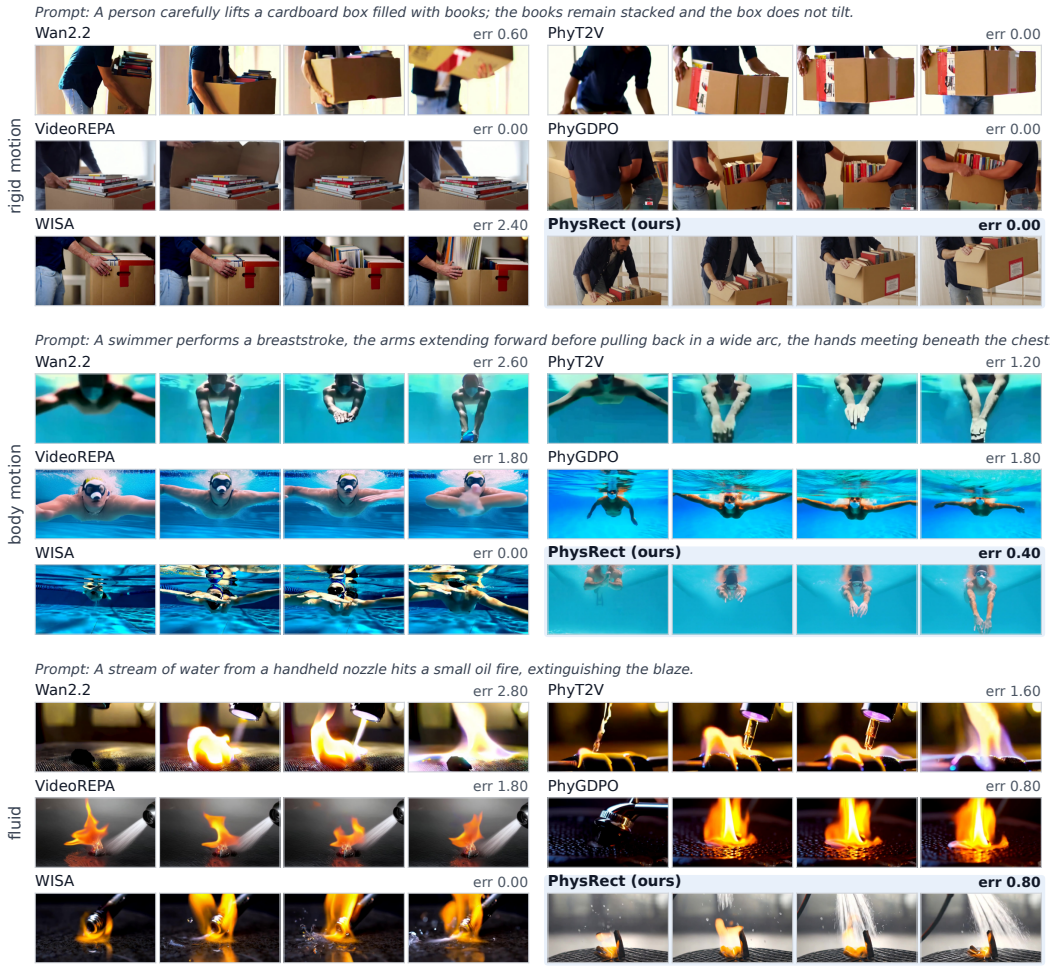


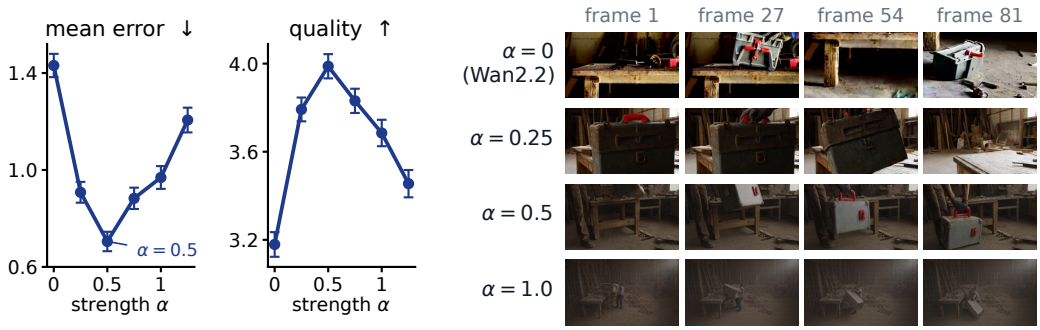
Figure 5: **Qualitative comparison.** Three prompts, one per mechanism, each shown for six methods at four frames, with the video's mean error next to the method name. The prompt is abridged.

through the full arc until the hands meet beneath the chest. The comparison with Wan2.2 shows the effect of the correction directly, since the two share the generator, the seed and the sampler and differ only in the text embedding: in the box scene (top), the books stay stacked as the box is lifted under PhysRect, while under Wan2.2 they shift and vanish. More comparisons are in Appendix C.2.

4.3 Ablation Study

We investigate the correction strength α, the architecture of the correction, and the form and weight of the separability term, on the same evaluation setting as Table 1.

Correction strength. Figure 6a sweeps α from 0 to 1.25 with the same experts, seeds and sampler. Error and quality follow the same trend, improving up to $\alpha = 0.5$ and degrading beyond it: the error falls from 1.43 at $\alpha = 0$, the unmodified Wan2.2, to 0.71, then rises again to 1.21 at $\alpha = 1.25$, and quality peaks at 3.99 at the same point. Figure 6b shows the sweep on one prompt. Notably, this optimum is larger than on the fitted embeddings themselves, where interpolating towards the fitted embedding reproduces the event already at $\alpha = 0.25$ and changes little beyond it (Appendix C.1). The experts approximate the fits rather than reproduce them, so a larger share of the predicted displacement is needed for the event to complete. At $\alpha = 0.5$ the toolbox leaves the bench and lands, with the framing, lighting and clutter of the uncorrected sample unchanged. At $\alpha = 1.0$ the displacement acts on the scene as well: the shot pulls back and darkens, consistent with a fitted embedding that also carries appearance, which a partial correction leaves out.



(a) (b)

Figure 6: **Correction strength.** (a) Mean error and quality against α on VideoPhy-2; error bars are standard errors of the mean over the 591 prompts. (b) One prompt, a heavy metal toolbox was taken down from the workbench: the uncorrected sample and three values of α at the same seed. At $\alpha = 0.5$ the toolbox leaves the bench and lands while the scene is kept; at $\alpha = 1.0$ the shot pulls back and darkens.

Table 2: **Architecture of the correction.** Same setting and columns as Table 1, at $\alpha = 0.5$.

Variant	Coherence	Material	Optics	Kinematics	Interaction	Mean ↓	Quality ↑
Wan2.2, no correction	1.52	1.03	0.74	1.96	1.91	1.431	3.18
Mixture of seven experts (ours)	**0.67**	**0.49**	**0.21**	**1.10**	**1.06**	**0.705**	**3.99**
Single network, no mechanisms	1.07	0.88	0.44	1.79	1.64	1.164	3.42
Row mean of the displacement, broadcast	1.45	1.01	0.58	1.80	1.77	1.322	3.35
Fixed mean displacement, no prompt dependence	1.42	0.99	0.65	1.77	1.76	1.317	3.36

Architecture of the correction. The correction of Eq. 5 is mechanism-specific, per token and prompt-dependent, and Table 2 removes each property in turn: one network of the same depth that receives no mechanism labels in place of the seven experts (33.6M parameters), the row mean of the predicted displacement broadcast to every row, and the mean displacement of the training fits, a fixed vector added to every prompt. Each removal sets the error back towards the backbone: the single network reaches 1.16, and the row mean and the fixed vector reach 1.32 against a backbone of 1.43, so the correction has to be mechanism-specific, per token and prompt-dependent to deliver its gain. In velocity space the fixed vector recovers little of what the experts do: on held-out events the distillation loss falls from 0.0429 without correction to 0.0414 with the fixed vector and to 0.0324 with the experts.

Form and strength of the separability term. The separability term $\mathcal{L}_{\text{sep}}$ keeps the fits of different mechanisms apart. Table 3 varies the form of the term, the prototype pull of Eq. 6 against a linear classifier head on $\bar{\Delta}$, and the gradient ratio γ. The prototype form is better at every α and γ, $\gamma = 0.1$ is better than 0.3 in five of the six comparisons, and $\alpha = 1.0$ is the worst column in every row, consistent with Fig. 6a.

Table 3: **Form and strength of the separability term.** Mean error on VideoPhy-2.

Form	γ	α=0.25	α=0.5	α=1.0
Linear head	0.1	1.035	1.004	1.063
Linear head	0.3	0.972	1.090	1.228
Prototype	0.1	**0.908**	**0.705**	**0.969**
Prototype	0.3	0.941	1.014	1.201

5 Conclusion

We presented PhysRect, which improves the physical consistency of a frozen video generator by rectifying its text embedding before sampling. A mixture of mechanism experts, distilled from embeddings fitted to real physical events, predicts the correction from the prompt alone, so the generator gains physical consistency at no cost per step and with its weights, sampler and guidance unchanged. On a physics benchmark it matches methods that retrain the generator at far lower cost.

Limitations. The fit reproduces the whole clip, so a fitted embedding carries appearance and camera along with physics, and the displacement the experts learn is not purely physical. The correction also steers the generator towards outcomes it can already produce and does not enforce physical law, so quantities such as acceleration or momentum stay unconstrained. We leave both to future work.

AI USE STATEMENT

In this work, we used generative AI tools for data annotation, data cleaning, and method implementation. We have not used generative AI tools for other required-disclosure tasks. Additionally, we used generative AI tools for polishing the writing and searching for related literature. We have reviewed all AI-assisted work: annotations and cleaned data were manually spot-checked by the authors, code was reviewed and tested for correctness, all cited references were verified by the authors, and all edited text was checked for accuracy. We take responsibility for the final content of this work, including text, claims or artifacts produced with the aid of generative AI.

REPRODUCIBILITY STATEMENT

To ensure that the proposed work is reproducible, Secs. 3.2 and 3.3 state the fitting objective, the distillation objective and the inference rule in closed form (Eqs. 5 to 9), and we have an explicit section (Appendix B) with implementation details, including the backbone and sampler settings, the two-stage fitting schedule, the expert architecture and training procedure, and the pipeline parameters. The evaluation (Sec. 4.1) uses the public VideoPhy-2 prompts and the public JudgeFit judge.

REFERENCES

Yogesh Balaji, Seungjun Nah, Xun Huang, Arash Vahdat, Jiaming Song, Qinsheng Zhang, Karsten Kreis, Miika Aittala, Timo Aila, Samuli Laine, Bryan Catanzaro, Tero Karras, and Ming-Yu Liu. eDiff-I: Text-to-image diffusion models with an ensemble of expert denoisers. *arXiv preprint arXiv:2211.01324*, 2022.

Hritik Bansal, Clark Peng, Yonatan Bitton, Roman Goldenberg, Aditya Grover, and Kai-Wei Chang. VideoPhy-2: A challenging action-centric physical commonsense evaluation in video generation. *arXiv preprint arXiv:2503.06800*, 2025.

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

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. *arXiv preprint arXiv:2512.24551*, 2025.

Yu Cao, Ziquan Liu, Zhensong Zhang, Jiankang Deng, Shaogang Gong, and Jifei Song. Each judge its own yardstick: Discovering per-vlm taxonomies for physical video evaluation. *arXiv preprint arXiv:2606.22918*, 2026.

Yuefan Cao, Chengyue Gong, Xiaoyu Li, Yingyu Liang, Zhizhou Sha, Zhenmei Shi, and Zhao Song. Richspace: Enriching text-to-video prompt space via text embedding interpolation. *arXiv preprint arXiv:2501.09982*, 2025.

Hila Chefer, Uriel Singer, Amit Zohar, Yuval Kirstain, Adam Polyak, Yaniv Taigman, Lior Wolf, and Shelly Sheynin. VideoJAM: Joint appearance-motion representations for enhanced motion generation in video models. In *International Conference on Machine Learning (ICML)*, 2025. arXiv:2502.02492.

Jooyoung Choi, Jungbeom Lee, Chaehun Shin, Sungwon Kim, Hyunwoo Kim, and Sungroh Yoon. Perception prioritized training of diffusion models. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2022. arXiv:2204.00227.

Kevin Clark, Paul Vicol, Kevin Swersky, and David J. Fleet. Directly fine-tuning diffusion models on differentiable rewards. In *International Conference on Learning Representations*, 2024.

Patrick Esser, Sumith Kulal, Andreas Blattmann, Rahim Entezari, Jonas Müller, Harry Saini, Yam Levi, Dominik Lorenz, Axel Sauer, Frederic Boesel, et al. Scaling rectified flow transformers for high-resolution image synthesis. In *Forty-first international conference on machine learning*, 2024.

Lin Geng Foo, Mark He Huang, Alexandros Lattas, Stylianos Moschoglou, Thabo Beeler, and Christian Theobalt. Physical simulator in-the-loop video generation. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2026. arXiv:2603.06408.

Rinon Gal, Yuval Alaluf, Yuval Atzmon, Or Patashnik, Amit H. Bermano, Gal Chechik, and Daniel Cohen-Or. An image is worth one word: Personalizing text-to-image generation using textual inversion. In *International Conference on Learning Representations (ICLR)*, 2023a. arXiv:2208.01618.

Rinon Gal, Moab Arar, Yuval Atzmon, Amit H. Bermano, Gal Chechik, and Daniel Cohen-Or. Encoder-based domain tuning for fast personalization of text-to-image models. *ACM Transactions on Graphics (SIGGRAPH)*, 2023b. arXiv:2302.12228.

Google DeepMind. Gemini 3 flash model card. `https://storage.googleapis.com/deepmind-media/Model-Cards/Gemini-3-Flash-Model-Card.pdf`, December 2025.

Yoav HaCohen, Nisan Chiprut, Benny Brazowski, Daniel Shalem, Dudu Moshe, Eitan Richardson, Eran Levin, Guy Shiran, Nir Zabari, Ori Gordon, et al. LTX-Video: Realtime video latent diffusion. *arXiv preprint arXiv:2501.00103*, 2024.

Yutong Hao, Chen Chen, Ajmal Saeed Mian, Chang Xu, and Daochang Liu. Enhancing physical plausibility in video generation by reasoning the implausibility. *arXiv preprint arXiv:2509.24702*, 2025.

Jonathan Ho and Tim Salimans. Classifier-free diffusion guidance. *arXiv preprint arXiv:2207.12598*, 2022.

Sangwon Jang, Taekyung Ki, Jaehyeong Jo, Saining Xie, Jaehong Yoon, and Sung Ju Hwang. Self-refining video sampling. *arXiv preprint arXiv:2601.18577*, 2026.

Bingyi Kang, Yang Yue, Rui Lu, Zhijie Lin, Yang Zhao, Kaixin Wang, Gao Huang, and Jiashi Feng. How far is video generation from world model: A physical law perspective. In *International Conference on Machine Learning (ICML)*, 2025. arXiv:2411.02385.

Diederik P. Kingma and Jimmy Ba. Adam: A method for stochastic optimization. In *International Conference on Learning Representations*, 2015.

Weijie Kong, Qi Tian, Zijian Zhang, Rox Min, Zuozhuo Dai, Jin Zhou, Jiangfeng Xiong, Xin Li, Bo Wu, Jianwei Zhang, et al. HunyuanVideo: A systematic framework for large video generative models. *arXiv preprint arXiv:2412.03603*, 2024.

Vincent Le Guen and Nicolas Thome. Disentangling physical dynamics from unknown factors for unsupervised video prediction. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2020.

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. *arXiv preprint arXiv:2503.09595*, 2025.

Yaron Lipman, Ricky T. Q. Chen, Heli Ben-Hamu, Maximilian Nickel, and Matt Le. Flow matching for generative modeling. In *International Conference on Learning Representations (ICLR)*, 2023. arXiv:2210.02747.

Jie Liu, Gongye Liu, Jiajun Liang, Ziyang Yuan, Xiaokun Liu, Mingwu Zheng, Xiele Wu, Qiulin Wang, Menghan Xia, Xintao Wang, et al. Improving video generation with human feedback. *Advances in Neural Information Processing Systems*, 38:82155–82192, 2026a.

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

Xingchao Liu, Chengyue Gong, and Qiang Liu. Flow straight and fast: Learning to generate and transfer data with rectified flow. In *International Conference on Learning Representations (ICLR)*, 2023. arXiv:2209.03003.

Ilya Loshchilov, Frank Hutter, et al. Fixing weight decay regularization in adam. *arXiv preprint arXiv:1711.05101*, 5(5):5, 2017.

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

Siwei Meng, Yawei Luo, Shu Zhang, and Ping Liu. When physical preferences meet semantic constraints: Physical and semantic direct preference optimization for text-to-video generation. *arXiv preprint arXiv:2607.16947*, 2026.

Saman Motamed, Laura Culp, Kevin Swersky, Priyank Jaini, and Robert Geirhos. Do generative video models learn physical principles from watching videos?, 2025. *URL https://arxiv.org/abs/2501.09038*, 1:2, 2025.

Byeonghu Na, Minsang Park, Gyuwon Sim, Donghyeok Shin, HeeSun Bae, Mina Kang, Se Jung Kwon, Wanmo Kang, and Il-Chul Moon. Diffusion adaptive text embedding for text-to-image diffusion models. *arXiv preprint arXiv:2510.23974*, 2025.

Hyelin Nam, Jaemin Kim, Dohun Lee, and Jong Chul Ye. Optical-flow guided prompt optimization for coherent video generation. *arXiv preprint arXiv:2411.15540*, 2024.

William Peebles and Saining Xie. Scalable diffusion models with transformers. In *IEEE/CVF International Conference on Computer Vision (ICCV)*, 2023. arXiv:2212.09748.

Joar Skalse, Nikolaus Howe, Dmitrii Krasheninnikov, and David Krueger. Defining and characterizing reward hacking. In *Advances in Neural Information Processing Systems*, 2022.

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.

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. *Advances in Neural Information Processing Systems*, 38:5388–5416, 2026a.

Peiyao Wang, Weining Wang, and Qi Li. Physcorr: Dual-reward dpo for physics-constrained text-to-video generation with automated preference selection. *arXiv preprint arXiv:2511.03997*, 2025.

Zixuan Wang, Yixin Hu, Wen Li, Feng Chen, Yan Liu, Duo Peng, and Yinjie Lei. Physically plausible video generation via visual-semantic chain-of-events conditioning. *arXiv preprint arXiv:2609.00656*, 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. *arXiv preprint arXiv:2603.09094*, 2026c.

Nicholas Watters, Andrea Tacchetti, Théophane Weber, Razvan Pascanu, Peter Battaglia, and Daniel Zoran. Visual interaction networks: Learning a physics simulator from video. In *Advances in Neural Information Processing Systems*, 2017.

Yuxiang Wei, Yabo Zhang, Zhilong Ji, Jinfeng Bai, Lei Zhang, and Wangmeng Zuo. ELITE: Encoding visual concepts into textual embeddings for customized text-to-image generation. In *IEEE/CVF International Conference on Computer Vision (ICCV)*, 2023. arXiv:2302.13848.

Jiajun Wu, Joseph J. Lim, Hongyi Zhang, Joshua B. Tenenbaum, and William T. Freeman. Physics 101: Learning physical object properties from unlabeled videos. In *British Machine Vision Conference*, 2016.

Qiyao Xue, Xiangyu Yin, Boyuan Yang, and Wei Gao. PhyT2V: LLM-guided iterative self-refinement for physics-grounded text-to-video generation. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2025. arXiv:2412.00596.

Xindi Yang, Baolu Li, Yiming Zhang, Zhenfei Yin, Lei Bai, Liqian Ma, Zhiyong Wang, Jianfei Cai, Tien-Tsin Wong, Huchuan Lu, and Xu Jia. VLIPP: Towards physically plausible video generation with vision and language informed physical prior. *arXiv preprint arXiv:2503.23368*, 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 *International Conference on Learning Representations*, 2025b.

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

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

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 (NeurIPS)*, 2025b. arXiv:2505.23656.

Qi Zhao, Xingyu Ni, Ziyu Wang, Feng Cheng, Ziyan Yang, Lu Jiang, and Bohan Wang. Synthetic video enhances physical fidelity in video synthesis. *arXiv preprint arXiv:2503.20822*, 2025.

A Appendix overview

This is the appendix for "Physics Before the First Step: Rectifying the Text Embedding of Frozen Video Generators". Tab. 4 summarizes the symbols used in the paper.

This appendix is organized as follows:

- Section B presents additional details of our approach.
- Section C presents additional qualitative and quantitative analysis.

B Additional details of our approach

Backbone. A prompt's embedding has L rows, one per token, 86 to 141 on our events with a median of 112. The transformer pads it to 512 rows internally. The VAE compresses by $r_t = 4$ and $r_s = 16$. Sampling uses UniPC with timestep shift 5, the generator's default.

Embedding fitting and Experts training. In the per-event fit, only C is optimized, the noise level is drawn as $\sigma = 5u/(1 + 4u)$ with $u \sim U(0, 1)$, the sampler's own schedule, and the loss is computed over the whole latent. The joint refit starts from these fits with the same optimizer and learning rate. One per-event fit takes about 0.48 GPU-hours. Each expert is preceded by a linear projection from d to 512 and followed by a zero-initialized projection back to d. Of the 600 fitted events, 576 are used for training and 24 are held out for validation. Training uses 300 warm-up steps, one event per GPU per micro-step and gradient accumulation of 2 on 4 GPUs, that is, 8 events per update. The distillation loss draws σ from the same schedule as the fit, and the gating weights are binary, $w_c = \mathbf{1}[\ell_c > 0]$.

Inference. Mechanism vectors are read from the prompt by Gemini 3 Flash (Google DeepMind, 2025), with $\ell_c = 1$ for the dominant mechanism, $\frac{1}{2}$ for one that is present and 0 otherwise.

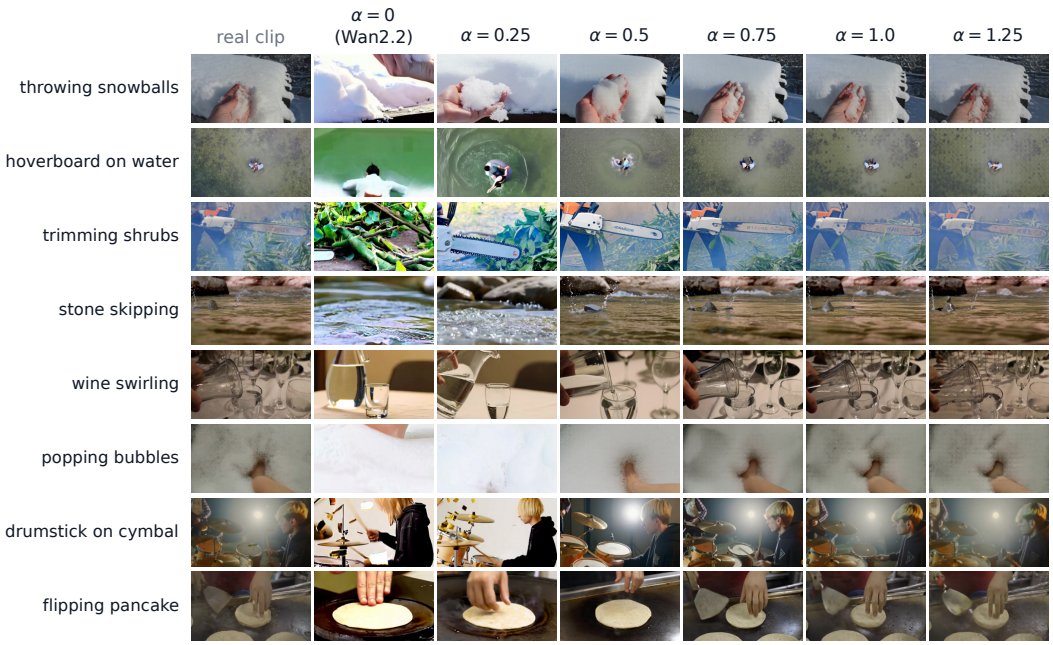


Figure 7: **Interpolating from the prompt embedding to the fitted one.**

Table 4: Symbols used in the paper.

Symbol	Meaning
Video and latent	
x	A video, $x \in \mathbb{R}^{T\times H\times W\times 3}$
T, H, W	Number of frames, height, width of a video
V	A source video
$\mathcal{P}(V)$	Intervals $[s_i, u_i]$ of V over which one physical process runs
a	First frame of a window $v_{a:a+T}$ of V
$\mathcal{E}, \mathcal{D}$	VAE encoder, VAE decoder
z	Latent of a clean video, $z = \mathcal{E}(x) \in \mathbb{R}^{t\times h\times w\times d_z}$
t, h, w, d_z	Latent frames, height, width, channels
r_t, r_s	Temporal and spatial compression ratios of the VAE
Flow matching and sampling	
σ	Noise level, $\sigma \in [0, 1]$; $\sigma = 1$ is pure noise
ϵ	Gaussian noise, $\epsilon \sim \mathcal{N}(0, I)$
z_σ	Noisy latent, $(1-\sigma)z + \sigma\epsilon$
v_θ	Velocity network with frozen parameters θ
$\mathcal{L}_{\text{FM}}$	Flow matching loss
S, σ_k	Number of sampling steps and their noise levels
$\hat{z}_\sigma$	Clean latent implied by the state at σ
$\hat{x}_\sigma$	Decoded clean estimate, $\mathcal{D}(\hat{z}_\sigma)$
g	Classifier-free guidance scale
Text embedding	
y	Text prompt
τ	Frozen text encoder
C	Text embedding, $C = \tau(y) \in \mathbb{R}^{L\times d}$
L, d	Number of prompt tokens (rows of C) and their dimension
N	Number of latent tokens
B	Number of transformer blocks
$W_k^{(b)}, W_v^{(b)}$	Key and value projections of block b
C_0	Embedding of the prompt as given
C^*	Embedding fitted to one event
f	Predictor from C_0 to its correction
α	Strength of the correction, $C_0 + \alpha f(C_0, \ell)$
Δ, Δ^*	Displacement $C - C_0$; the fitted one, $C^* - C_0$
$\bar{\Delta}$	Row mean of Δ
ℓ	Mechanism vector of an event or prompt, $\ell \in [0, 1]^K$
c^*	Dominant mechanism, $\arg\max_c \ell_c$
$\mathcal{L}_{\text{sep}}$	Separability term of the fit, $\|\bar{\Delta} - \mu_{c^*}\|^2 - \|\bar{\Delta} - \mu_{c'}\|^2$
λ, γ	Weight of the separability term; its gradient ratio
f_c, w_c	Expert for mechanism c and its weight
$\mathcal{L}_f$	Distillation loss of the experts
Physical events	
e	A physical event, $e = (x, m, y)$, a window of V holding one process
K	Number of physical mechanisms, $K = 7$: rigid motion, body motion, fluid, granular, deformation, fracture, appearance
m	Mask over the moving subject in every frame

C ADDITIONAL QUALITATIVE AND QUANTITATIVE ANALYSIS

C.1 INTERPOLATING TOWARDS THE FITTED EMBEDDING

Before distilling the fits into experts, we explored how much of a fitted displacement the generator needs. Figure 7 interpolates from the prompt embedding C_0 to the fitted embedding C^* on eight events, with the seed and the sampler fixed. At $\alpha = 0.25$ the described action already takes place, the snowball is thrown, the stone skips and the pancake is flipped, while the scene, the framing and the subject keep the appearance the generator gave them under C_0, so the video differs from the real clip. From $\alpha = 0.5$ the generator reproduces the real clip, and $\alpha = 0.75$, 1.0 and 1.25 change little beyond it, with grain appearing at 1.25. A quarter of the fitted displacement therefore carries the physics of the event, and the remainder carries its appearance.

C.2 MORE QUALITATIVE COMPARISONS

Figures 8 and 9 show ten more prompts in the layout of Fig. 5. The pattern of Sec. 4.2 holds across them: PhysRect keeps the subject intact and carries the described process to its outcome on most prompts, while the other methods more often depart from it. Prompts on which PhysRect scores worse than Wan2.2 are included, so that the selection is not limited to its successe.

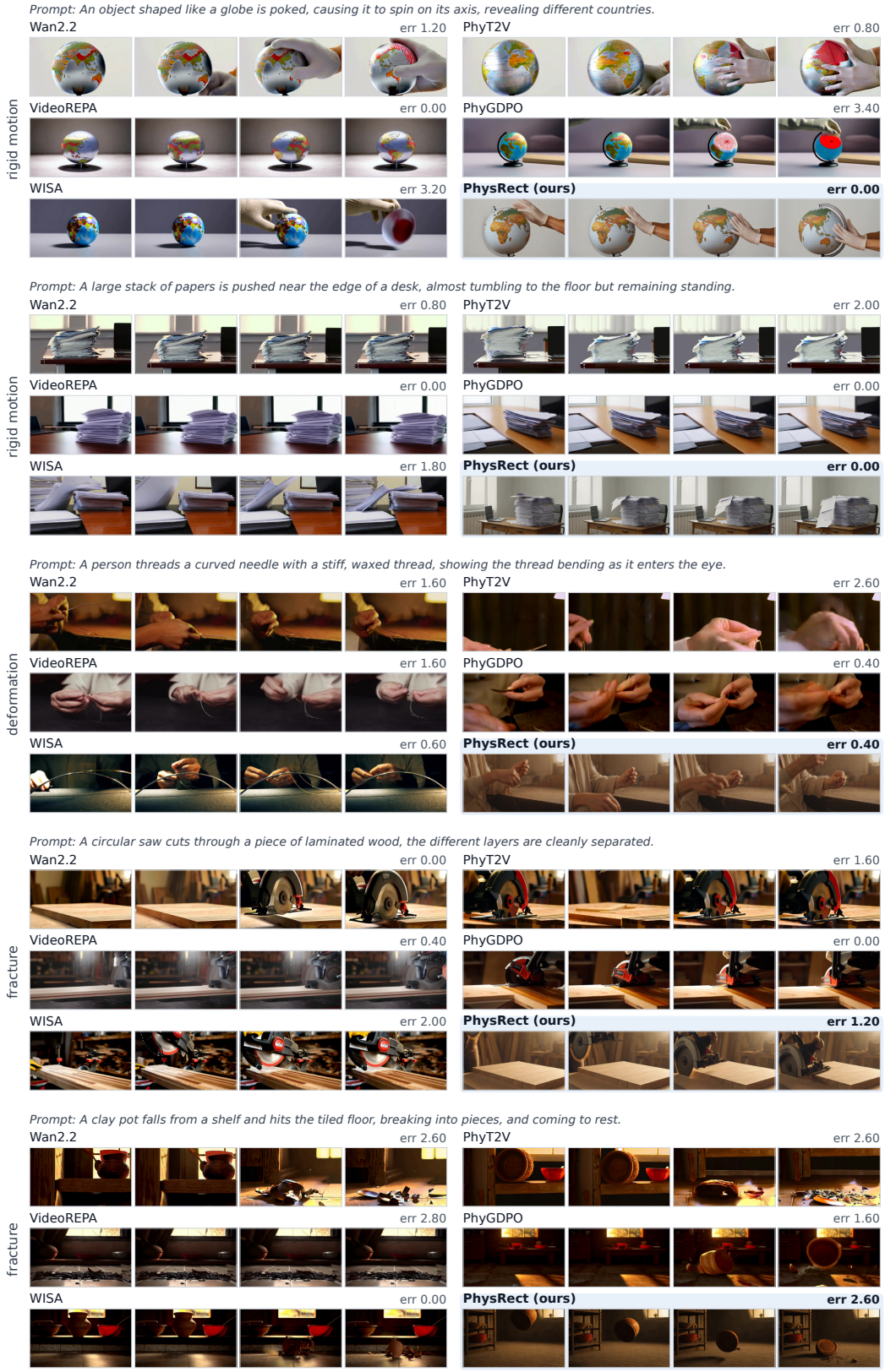


Figure 8: **Further qualitative comparisons.** Layout as in Fig. 5. The prompt is abridged.



Figure 9: **Further qualitative comparisons.** Layout as in Fig. 5. The prompt is abridged.