Fluid-Gen-Zero: Grounding Pretrained Video Generators in Physics without Training

Anonymous Authors[1]

Abstract

We present **Fluid-Gen-Zero**, a training-free framework for physics-aware fluid video generation that elegantly decouples physical reasoning from appearance synthesis. Our key insight is to delegate motion dynamics to a physics simulator while preserving the appearance modeling capacity of pretrained video generators. We bridge these two domains through a two-level agentic workflow: *generation-time planning*, where a rule-grounded VLM agent parses intent, observes simulation evidence, and organizes generation into physically feasible clips, and *latent-space guidance*, which injects simulation-derived signals into the denoising process via region-aware latent wrapping. This plug-and-play design is compatible with the latest video models. We further introduce a benchmark for evaluating fluid-object interactions. Across Tora (CogVideoX-based) and VACE (Wan-based) backbones, Fluid-Gen-Zero reduces object trajectory error by 30.0% to 48.0%, fluid trajectory error by up to 21.1%, and receives 62.86% to 87.14% overall human preference over baselines. Code and data will be released upon acceptance.

1. Introduction

Video foundation models (VFMs) achieve remarkable visual fidelity (Agarwal et al., 2025; Brooks et al., 2024; DeepMind, 2025; Kuaishou, 2024; Team, 2025; Yang et al., 2025), yet they are not inherently grounded in physical laws. This challenge is particularly evident in fluid-object interactions, where complex, deformable dynamics cannot be reliably inferred from text or standard diffusion priors (Bridson, 2015; Hu et al., 2018; Stam, 2023). Consequently, state-of-the-art generators frequently produce visually appealing but physically implausible motion (Bansal et al., 2024; Meng et al., 2024; Wang et al., 2025b).

[1]Anonymous Institution, Anonymous City, Anonymous Region, Anonymous Country. Correspondence to: Anonymous Author <anon.email@domain.com>.

Preliminary work. Under review by the AI4Physics Workshop at the International Conference on Machine Learning (AI4Physics@ICML 2026). Do not distribute.

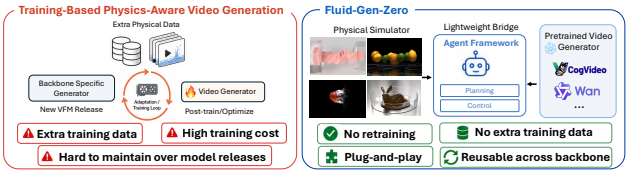

Figure 1. **Training-based *vs.* Training-free physics-aware video generation.** *(Left)* Existing approaches rely on architecture-coupled post-training to inject physical priors, creating a rigid dependency on specific model backbones. *(Right)* **Fluid-Gen-Zero** decouples physical reasoning from appearance synthesis. By introducing physics awareness entirely at inference time, it offers a plug-and-play solution compatible with rapidly evolving video foundation models.

To bridge this gap, recent works inject physical priors through fine-tuning (Bansal et al., 2024; Li et al., 2025a; Meng et al., 2024; Wang et al., 2025b), preference optimization (Cai et al., 2025; Chen et al., 2025b; Xue et al., 2025), or auxiliary objectives (Chen et al., 2025a; Foo et al., 2026; Li et al., 2025d; Liu et al., 2026; Xie et al., 2025; Zhan et al., 2026). While these methods improve physical grounding, their reliance on architecture-coupled post-training presents scalability challenges. First, updating the weights of massive video generators incurs prohibitive computational costs. Second, it involves an inherent trade-off: optimizing for physical constraints frequently degrades the generator's native perceptual quality and prompt-adherence (Zhou et al., 2026). More importantly, as VFMs scale and evolve at an unprecedented pace, maintaining architecture-coupled post-training paradigms becomes increasingly unsustainable. An architecture-specific physics adapter trained today risks immediate obsolescence when a new VFM is released tomorrow—a dynamic reminiscent of the "bitter lesson" (Sutton, 2019).

This highlights the critical necessity of a **training-free** approach. By treating physics awareness as an inference-time capability, a training-free framework avoids costly retraining, preserves the base model's visual fidelity, and offers

plug-and-play compatibility across future architectures. A sustainable solution must therefore **decouple** physical reasoning from the generator's weights entirely, prompting a fundamental question:

> *How to bridge pretrained video generators and physics-awareness without training?*

In this work, we propose **Fluid-Gen-Zero**, a training-free framework that decouples physical reasoning from appearance synthesis, enabling plug-and-play physics awareness across production-level video generators in fluid scenes. Our intervention is logically based on two universally accessible handles: the generation timeline and the internal denoising trajectory.

Fluid-Gen-Zero leverages these handles at two complementary levels. At the **generation-planning level**, a VLM agent translates input intent and simulation evidence into adaptive clip plans, utilizing prompt refinements and protected no-cut intervals. At the **latent-control level**, simulation-derived signals are injected into the denoising process via time-dependent, region-aware latent wrapping. Together, these mechanisms recast physics awareness from a rigid training objective into a portable, inference-time capability. We summarize our contributions below:

- We propose **Fluid-Gen-Zero**, a training-free framework that introduces physics awareness through generation-planning and latent-control, providing plug-and-play compatibility with multiple pretrained video generators.
- We contribute a benchmark of 50 curated fluid and fluid-object samples featuring 401-frame rollouts, dense object/fluid trajectories, masks, and optical flow across 10 fluid-phenomenon categories.
- Across Tora (CogVideoX-based) and VACE (Wan-based) backbones, Fluid-Gen-Zero reduces object trajectory error by 30.0% to 48.0%, decreases fluid trajectory error by up to 21.1%, and achieves 62.86% to 87.14% overall human preference over baselines.

2. Related Work

Physics-based simulation for fluids and interactions. Physics-based simulation provides principled tools for modeling fluid dynamics and multi-material coupling. Classical methods (SPH/PBF, MPM) (Bridson, 2015; Hu et al., 2018; Monaghan, 1992; Stam, 2023; Stomakhin et al., 2013) and modern engines (PyBullet, MuJoCo, Taichi, Warp, Genesis) (Authors, 2024; Hu et al., 2019; Macklin, 2022; Todorov et al., 2012) accurately model state evolution while preserving causality, contact, and force propagation. While these simulators excel at modeling physical dynamics, their reliance on explicit parameterization and predefined geometry creates an inherent trade-off, as they do not account for the open-vocabulary appearance, fine-grained textures, and complex lighting inherent to generative visual models. Motivated by these complementary strengths, our approach bridges the physical reliability of simulation with the realistic appearance synthesis of pretrained video generators.

Controllable video generation. Recent video foundation models have achieved rapid progress in visual quality and temporal coherence (Agarwal et al., 2025; Brooks et al., 2024; DeepMind, 2025; Kuaishou, 2024; Team, 2025; Yang et al., 2025). To increase controllability, prior works introduce spatial or motion conditions such as masks, depth, drag operations, and trajectories (Li et al., 2025c; Ling et al., 2024; Namekata et al., 2024; Weijia Wu, 2024). Methods like MotionCtrl (Wang et al., 2024) and Tora (Zhang et al., 2025) enable precise motion control, while recent systems (MagicMotion (Li et al., 2025b), WanMove (Chu et al., 2025), ATI (Wang et al., 2025a), Time-to-Move (Singer et al., 2025)) extend this through dense representations or training-free schedules. While these methods excel at 2D screen or latent space control, generating physically valid 3D dynamics remains an open challenge, particularly for fluids governed by complex factors like viscosity and contact. Building on these insights, we incorporate physical simulation to provide structurally grounded 3D motion priors, complementing trajectory-based controls with physics-aware synthesis.

Physics-informed generative models. A growing line of work aims to improve physical consistency in generative video models. Approaches range from supervised fine-tuning and preference optimization (Bansal et al., 2024; Cai et al., 2025; Chen et al., 2025b; Li et al., 2025a; Meng et al., 2024; Wang et al., 2025b) to inference-time reasoning and simulator-guided generation (Foo et al., 2026; Xue et al., 2025). Recent hybrid systems deeply couple simulation with generation: WonderPlay (Li et al., 2025d) optimizes explicit 4D representations, PSIVG (Foo et al., 2026) employs test-time optimization on reconstructed templates, while RealWonder (Liu et al., 2026) and PerpetualWonder (Zhan et al., 2026) leverage simulation to condition distilled or long-horizon generators. Related works (PhysGen3D (Chen et al., 2025a), PhysAnimator (Xie et al., 2025)) similarly utilize 3D simulation for animation. While these methods demonstrate the value of physical simulation, a key design choice in many such pipelines is their reliance on model-specific training, test-time optimization, or explicit representation building. Complementary to these directions, Fluid-Gen-Zero provides a plug-and-play, training-free interface for standard video generators without updating weights or optimizing scene representations.

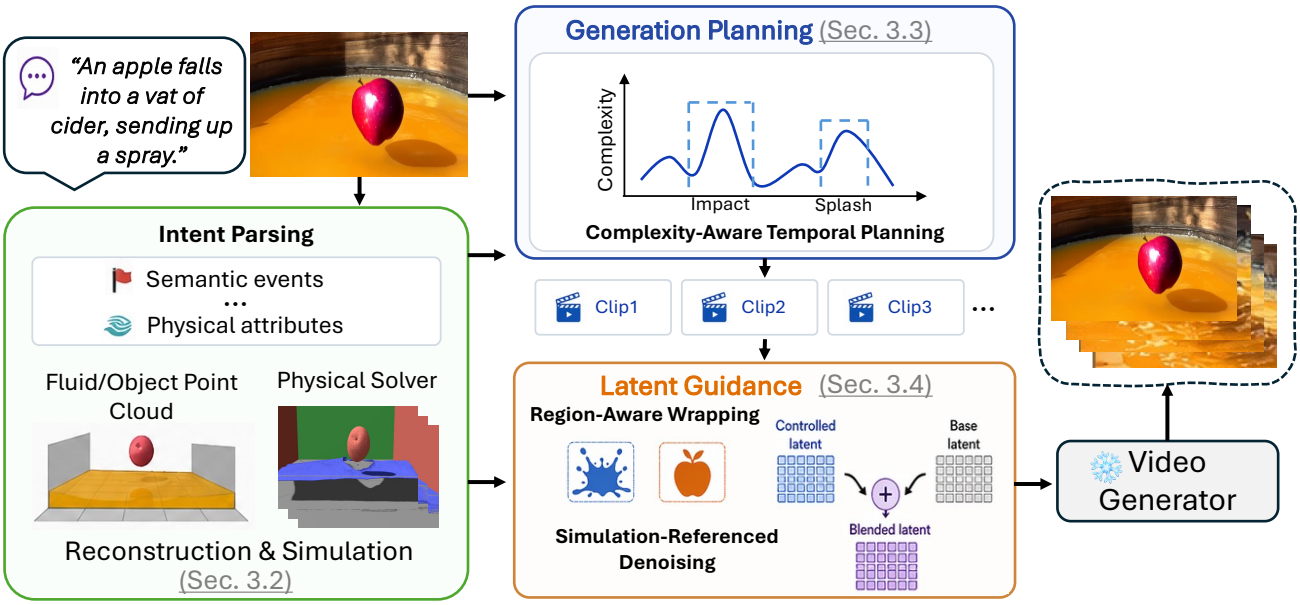

Figure 2. **Fluid-Gen-Zero pipeline.** Starting from an input image and text prompt, Fluid-Gen-Zero reconstructs a simulatable 3D scene to roll out fluid-object dynamics. The resulting simulation evidence then guides a frozen video foundation model (VFM) via a two-level workflow: *generation-level planning* and *latent-level control*. This design gracefully decouples physical reasoning from appearance synthesis, enabling plug-and-play physics awareness while preserving the generator's native visual fidelity.

3. Method

3.1. Overview

Fluid-Gen-Zero is a training-free image-to-video framework for fluid and fluid-object generation. Given an input image I_0, a prompt p, and a frozen pretrained video generator G_θ, our goal is to synthesize a prompt-faithful and visually realistic video that follows a plausible physical rollout. As illustrated in Figure 2, we decouple physical reasoning from appearance synthesis: a simulator provides structured dynamics such as trajectories and optical flow, while the pretrained generator preserves photorealistic appearance. Fluid-Gen-Zero first parses the input and reconstructs a simulator-ready scene to obtain a simulation bundle $\mathcal{S}$, plans generation clips from $\mathcal{S}$, and finally injects simulation-derived references into selected latent regions and denoising steps of G_θ.

3.2. Simulation-Grounded Scene Representation

This stage converts the input image and prompt into a scene state that can be used by both the simulator and the video generator, so later stages can plan clip boundaries and guide denoising with the same physical rollout. We construct a shared intermediate representation that bridges physical simulation and video generation by storing simulator-side quantities, such as 3D point clouds and physical parameters, together with generator-side image-space signals, such as projected masks, optical flow, and reference videos.

Fluid-Gen-Zero builds this representation in three steps. First, it parses the image and prompt into an intent record $\mathcal{I}$. A rule-based parser maps common motion and material words to valid simulator parameters, while a VLM resolves image-grounded information such as object category, scene context and event tags. Second, Fluid-Gen-Zero reconstructs a simulation-ready scene from the input image. It segments I_0 into object, fluid, and background regions (Liu et al., 2023; Ravi et al., 2024), estimates depth (Lin et al., 2025), lifts visible regions into 3D point cloud, and reconstructs and registers the foreground object (Bowen Wen, 2024; Xiang et al., 2024). This gives the initial object point cloud X_0^o and fluid point cloud X_0^f, with the background kept as static context. Finally, Fluid-Gen-Zero runs the physics simulator using the reconstructed scene and parsed parameters. The coupled object-fluid state is advanced by

$$(X_{t+1}^o, X_{t+1}^f) = \Phi_{\Theta_{\text{sim}}}(X_t^o, X_t^f), \quad t = 0, \ldots, T_{\text{sim}} - 1 \tag{1}$$

where T_{sim} is the simulation horizon, $\Phi_{\Theta_{\text{sim}}}$ denotes one simulation step, and Θ_{sim} collects simulator parameters parsed from $\mathcal{I}$, such as material properties and gravity. The simulated trajectories are projected back to the image plane and resampled to the video generation horizon of T frames. The final simulation bundle $\mathcal{S} = \{X_{1:T}^o, X_{1:T}^f, M_{1:T}^o, M_{1:T}^f, F_{1:T-1}, R_{1:T}^{\text{sim}}, V_{1:T}^{\text{sim}}\}$ contains object/fluid trajectories, projected masks, optical flow, a rendered simulation preview, and an appearance-preserving reference video, respectively. Additional details are provided in Appendix A and Appendix D.

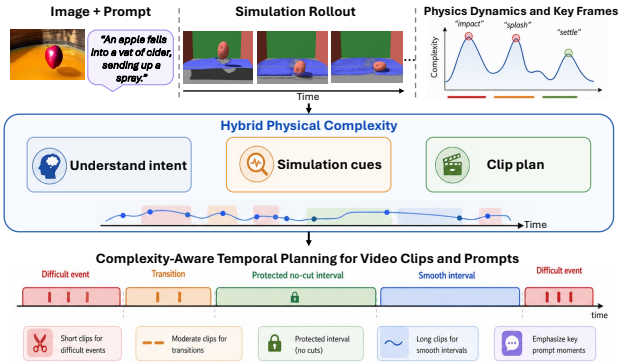

Figure 3. **Long-horizon simulation-to-video planning.**

3.3. Planning Long-Horizon Simulation-to-Video Generation

Generation planning turns one simulated physical rollout into a sequence of video-generation clips that can be stitched into a long-horizon, physically coherent video. Pretrained video generators usually operate on fixed-length clips, but the physical complexity of fluid motion is highly uneven over time. Smooth fluid motion can tolerate longer temporal spans, but abrupt events such as splashes and occlusions should be kept within the same clip rather than split by clip boundaries. So given the prompt p, the intent record $\mathcal{I}$, and the simulation bundle $\mathcal{S}$, Fluid-Gen-Zero computes a simulation-derived complexity curve, protects important physical events from being cut, and outputs a K-clip plan $\mathcal{B} = \{(s_i, e_i, p_i)\}_{i=1}^{K}$. Each tuple contains the start frame s_i, end frame e_i, and prompt p_i of clip i. Figure 3 illustrates this workflow, and Figure 4 shows a representative planning result.

3.3.1. Hybrid Physical Complexity via Physics Simulation and VLM Reasoning

Fluid-Gen-Zero first computes a complexity score for each simulation step, indicating where a frozen video generator is most likely to drift from the physical rollout. Some intervals mainly contain smooth motion, while others might contain contact changes that are more difficult to generate consistently. To capture this difference, Fluid-Gen-Zero uses $\mathcal{S}$ and $\mathcal{I}$ to produce a scalar complexity curve C_t over time. Simulation metrics determine when physical changes occur and how strong they are, while the VLM is restricted to semantic interpretation of the prompt and simulation preview, such as whether a contact or occlusion is especially important for the video. This keeps planning grounded in measurable rollout signals while still accounting for visually meaningful events. Fluid-Gen-Zero computes the hybrid score as

$$C_t = w_p C_t^{\text{phys}} + w_v C_t^{\text{vis}} + w_s C_t^{\text{sem}}, \tag{2}$$

where C_t^{phys} measures 3D simulation dynamics such as object speed, acceleration, fluid spread, and object-fluid contact; C_t^{vis} measures projected image-space difficulty such as large optical flow and fast mask-area change; and C_t^{sem} increases near prompt-relevant events, such as contact, splash, swirl, or occlusion, by localizing these events on the simulation timeline with event tags and VLM-estimated importance. All components are robustly normalized before fusion; the weights w_p, w_v, w_s are final normalized weights obtained by combining the VLM-predicted continuous weight with a fixed default weight. For example, pouring water into a glass emphasizes contact and fluid-boundary changes, while a floating-object scene emphasizes object motion and stability. More details are provided in Appendix B.

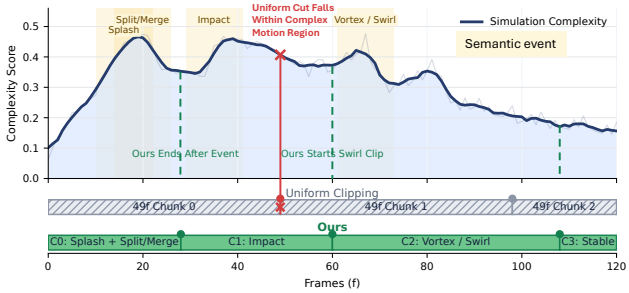

Figure 4. A representative Fluid-Gen-Zero planning output shows how simulation-derived complexity is translated into adaptive generation clips.

3.3.2. Complexity-Aware Temporal Planning for Video Clips and Prompts

After computing C_t, temporal planning converts the complexity curve into a discrete clip plan that a video generator can actually generate. It will produce the clip plan $\mathcal{B} = \{(s_i, e_i, p_i)\}_{i=1}^{K}$ from C_t, the protected event intervals and the original prompt p. The key intuition is that not all parts of a physical rollout should be generated with the same temporal span. High-complexity intervals are covered by shorter clips to keep difficult physical events generated in one model forward without any unrolling, while low-complexity intervals can use longer clips to reduce unnecessary stitching. We use a simple temporal planner to select clip boundaries from the simulation-derived complexity curve. A boundary b denotes a frame index. Starting from the current clip start s, the planner enumerates backbone-valid candidate end frames, scores each candidate boundary, selects the lowest-scoring one, and then proceeds greedily from left to right. Adjacent clips share the boundary frame as a rolling first-frame anchor, and the duplicated anchor frame is removed after stitching. Protected no-cut intervals are built around high-complexity scores and important semantic events, so the planner avoids splitting unstable physical events across clips. For a candidate boundary b, we use the following score:

$$\mathcal{J}(b) = \lambda_c \bar{C}_{\mathcal{N}(b)} + \lambda_l |L(b) - L^\star| - \lambda_e B_{\text{event}}(b) + \lambda_f \mathbb{1}[b \in \mathcal{F}]. \tag{3}$$

Each term captures a complementary requirement for a good clip boundary. $\mathcal{N}(b)$ is a small temporal neighborhood around b, and $\bar{C}_{\mathcal{N}(b)}$ is the average complexity in that neighborhood, discouraging cuts in locally unstable regions. $L(b)$ is the clip length induced by ending the current clip at b, and $L^\star$ is the preferred length for the current interval, keeping the selected clip length compatible with the generator while adapting to the local complexity level. $B_{\text{event}}(b)$ gives a bonus to boundaries near the start or end of a meaningful event, where a cut is more visually natural. $\mathcal{F}$ denotes the set of protected no-cut frames, and the indicator term strongly penalizes candidates that split high-complexity or semantically important events. The coefficients $\lambda_c, \lambda_l, \lambda_e, \lambda_f \geq 0$ balance these four factors.

In practice, the planner evaluates $\mathcal{J}(b)$ over backbone-valid candidate boundaries and selects the lowest-scoring boundary at each greedy step. Thus, minimizing $\mathcal{J}(b)$ places boundaries in low-risk regions or near natural event transitions, while keeping unstable physical events inside the same generated clip. After temporal boundaries are fixed, Fluid-Gen-Zero constructs a prompt p_i for each clip by preserving the global scene description and adding only the local physical event inside that interval, such as stable object-fluid contact or detailed splash formation. Full implementation details are provided in Appendix B.

3.4. Physics-Induced Latent-Space Guidance

Generation planning decides when to clip the unrolling and how to revise the prompt, but it still cannot align motions (generated in each video clip) with the physics simulation. Latent-space guidance bridges this gap by using the simulation reference video and object/fluid masks to guide denoising. The reference video provides coarse motion and layout, while the pretrained generator remains responsible for texture, lighting, and other visual details. We encode the reference into latent space, initialize sampling from a noised reference, and apply region-aware wrapping during denoising. We separate object and fluid regions because they require different types of control: objects should preserve identity and shape, whereas fluids are deformable and need more freedom for appearance synthesis. As shown in Figure 5, object regions are anchored more strongly, while fluid regions use softer guidance so the generator can still synthesize realistic liquid appearance. Details are provided in Appendix C.

Simulation-Referenced Denoising. Latent guidance starts denoising near the simulated motion rather than from an unconstrained latent state. For readability, we omit the clip index i here. For each clip, the simulation reference video V^{sim} is encoded by the VAE as z^{sim}. The object M^o and fluid masks M^f are projected to the latent resolution. Object support is removed from the fluid mask by $M^{f\setminus o} = M^f(1 - M^o)$. Instead of starting from pure Gaussian noise, Fluid-Gen-Zero uses simulation-guided noise. With $\epsilon \sim \mathcal{N}(0, I)$ and scheduler noise level σ_k at denoising step k, we define $r_k^{\text{sim}} = (1 - \sigma_k)z^{\text{sim}} + \sigma_k \epsilon$ and set the initial latent as $z_{k_0} = r_{k_0}^{\text{sim}}$. This places the denoising trajectory near the simulated layout and motion from the beginning, while the remaining noise still leaves room for the video model to synthesize realistic appearance. This is similar in spirit to partial noising for image editing (Meng et al., 2021; Singer et al., 2025), but here it is applied to video latents and later combined with region-aware guidance.

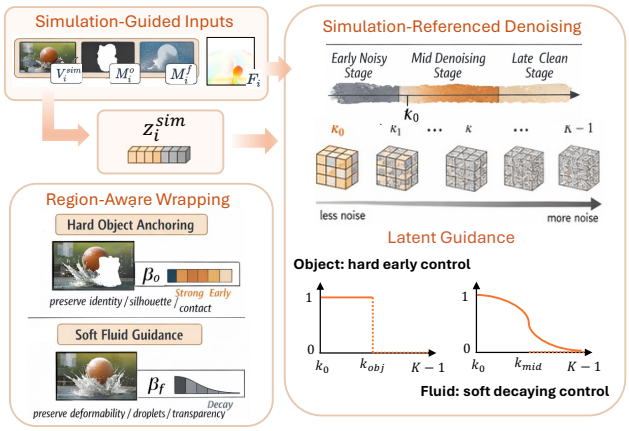

Figure 5. **Physics-induced latent-space guidance.** Fluid-Gen-Zero injects simulation-derived guidance into the denoising process of a frozen pretrained video generator without retraining.

Region-Aware Latent Wrapping. After initialization, the simulation rollout should constrain different regions during denoising because objects and fluids play different roles in the generated video. Objects should keep identity, shape, and contact geometry, while fluids should follow the simulated motion without losing fine visual detail. At each denoising step, the frozen generator first updates the current latent using its native denoising rule conditioned on the clip prompt p and backbone condition $\mathcal{C}$, such as optical flow, source masks, or first-frame guidance. We denote the updated latent by $\tilde{z}_{k+1}$. Fluid-Gen-Zero then wraps this update with the noised simulation reference in selected regions. For object regions, at the denoising step k, we use strong early anchoring:

$$z_{k+1}^o = (1 - \beta_k^o M^o) \odot \tilde{z}_{k+1} + \beta_k^o M^o \odot r_{k+1}^{\text{sim}}, \quad (4)$$

where $\beta_k^o \in \{0, 1\}$ is a binary indicator. We use an early schedule $\beta_k^o = \mathbb{1}[k_0 \leq k < k_o]$, so within $[k_0, k_o)$ the object region is fully replaced by the simulation reference, after which the generator's own update is preserved. This forces the object to follow the simulated position and silhouette while the global structure is being formed. Here z_{k+1}^o is the intermediate latent after object-region guidance, and k_o is the endpoint of the object control denoising step. For fluid regions, we use a softer schedule:

$$z_{k+1} = (1 - \beta_k^f M^{f\setminus o}) \odot z_{k+1}^o + \beta_k^f M^{f\setminus o} \odot r_{k+1}^{\text{sim}}, \quad (5)$$

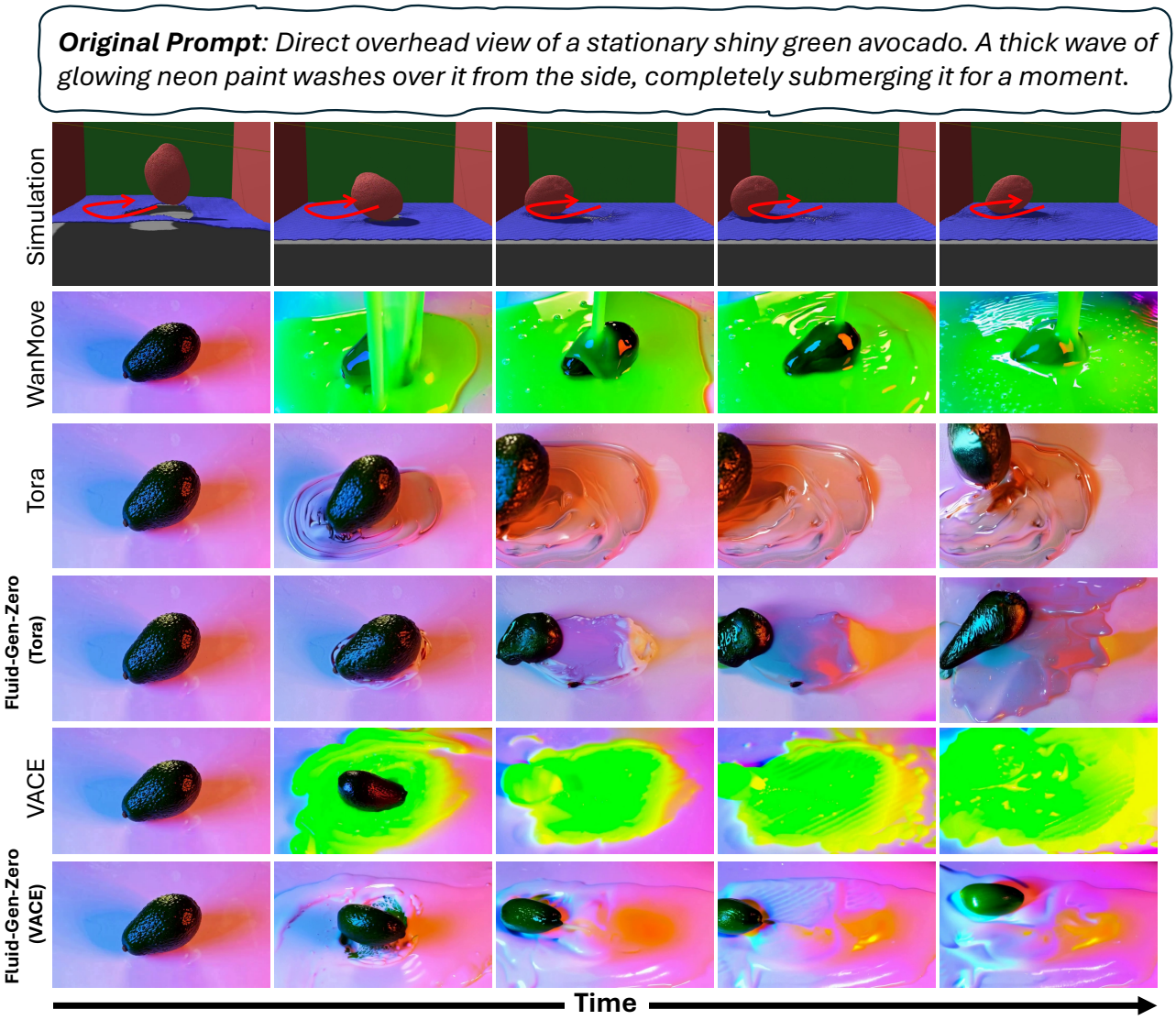

Figure 6. **Qualitative comparison on fluid-object interaction.** We compare simulation, WanMove, Tora, VACE, and Fluid-Gen-Zero variants over sampled frames. Each Fluid-Gen-Zero clip uses the corresponding refined clip-level prompt based on the original prompt.

with $\beta_k^f = \alpha_0 \max(0, 1 - (k - k_0)/(k_f - k_0))\mathbb{1}[k_0 \leq k < k_f]$. Unlike the binary β_k^o, $\beta_k^f \in [0, \alpha_0]$ continuously blends the simulation reference with the generator's update and decays linearly from α_0 to 0. Here z_{k+1} is the final latent after both object and fluid guidance, α_0 is the maximum fluid guidance strength, and k_f is the endpoint of the fluid control denoising step. The fluid guidance is stronger at early denoising steps, when the generator mainly determines the fluid region and motion direction, and then decays so the generator can refine liquid appearance. This asymmetric wrapping keeps objects aligned with the simulation, preserves the global motion of fluids, and avoids over-constraining visual effects such as transparency and droplets.

4. Experiments

4.1. Experimental Setup and Evaluation Protocol

We evaluate Fluid-Gen-Zero on a simulation-driven benchmark of 50 fluid and fluid-object interaction samples, with 40 held-out samples for testing. Unless otherwise specified, we use 401-frame physical rollouts for long-horizon generation; Table 3 uses 121-frame rollouts and each backbone's native video length to isolate latent control. We compare against representative pretrained and motion-conditioned video generators, including WanMove (Chu et al., 2025), Tora (Zhang et al., 2025) (CogVideoX-based), and VACE (Jiang et al., 2025) (Wan-based). For Tora and VACE, we additionally evaluate planning-only, latent-control-only, and complete Fluid-Gen-Zero variants. More

Table 1. **Main results on fluid and fluid-object interaction generation.** We compare the complete **Fluid-Gen-Zero** system against representative baselines.

Method	Size	Obj. Traj. Err. ↓	Flu. Traj. Err. ↓	Subj. Cons. ↑	Temp. Flick. ↑	Aesth. Qual. ↑	Img. Qual. ↑	Physical Commonsense ↑
WanMove	14B	121.14	110.47	0.829	0.967	0.520	0.667	0.134
Tora	5B	107.46	122.61	0.796	0.980	**0.470**	0.657	0.123
Fluid-Gen-Zero (Tora)	5B	**55.92**	**96.78**	**0.901**	**0.995**	0.462	**0.668**	**0.177**
VACE	1.3B	62.43	127.21	0.708	0.988	0.410	0.482	0.115
Fluid-Gen-Zero (VACE)	1.3B	**43.68**	**124.00**	**0.787**	**0.993**	**0.437**	**0.590**	**0.118**

Table 2. **Effectiveness of complexity-aware planning on multiple backbones.** We compare backbone baselines with **uniform clip segmentation** against variants with generation-time planning.

Method	Size	Obj. Traj. Err. ↓	Flu. Traj. Err. ↓	Subj. Cons. ↑	Temp. Flick. ↑	Aesth. Qual. ↑	Img. Qual. ↑	Physical Commonsense ↑
WanMove	14B	121.14	110.47	0.829	0.967	0.520	0.667	0.134
Tora	5B	**107.46**	122.61	0.796	0.980	0.470	0.657	0.123
Tora w/ Planning	5B	121.35	**102.20**	**0.798**	0.980	**0.475**	**0.662**	**0.138**
VACE	1.3B	62.43	**127.21**	0.708	**0.988**	0.410	0.482	0.115
VACE w/ Planning	1.3B	**56.73**	134.00	**0.718**	0.987	**0.436**	**0.540**	0.115

implementation and benchmark details are provided in Appendix D and E.

Metrics. We evaluate simulation-aligned controllability with object and fluid trajectory errors, visual and temporal quality with VBench subject consistency, temporal flickering, aesthetic quality, and imaging quality (Huang et al., 2024), and physical plausibility with VideoPhy physical commonsense (Bansal et al., 2024). Since trajectory errors and perceptual metrics cannot fully capture fluid appearance, boundary coherence, and human-perceived realism, we further provide a human preference study. Additional results are provided in Appendix F.

4.2. Main Results on Fluid and Fluid-Object Generation

Table 1 reports the main comparison on Tora and VACE backbones, with WanMove included as a strong large-scale reference model. Across both backbones, Fluid-Gen-Zero consistently improves simulation-aligned controllability, temporal stability, perceptual quality, and physical commonsense over the corresponding base generators. On Tora, Fluid-Gen-Zero reduces object trajectory error from 107.46 to 55.92 and fluid trajectory error from 122.61 to 96.78, while also improving subject consistency, temporal flickering, imaging quality, and physical commonsense. On VACE, it improves all reported metrics, reducing object trajectory error from 62.43 to 43.68 and increasing imaging quality from 0.482 to 0.590. Notably, despite using much smaller backbones than the 14B WanMove model, Fluid-Gen-Zero remains competitive or better on several alignment and quality metrics, suggesting that explicit simulation-grounded planning and control can compensate for part of the gap in native model capacity. Figure 6 provides representative qualitative comparisons.

4.3. Effectiveness of Complexity-Aware Planning

We next isolate generation-time planning by comparing uniform clip generation with complexity-aware planning while keeping latent-space guidance disabled. Figure 7 shows a representative rollout where the high-complexity region predicted from simulation is covered by the VLM-suggested no-cut interval and preserved within the final clip plan. The quantitative results in Table 2 further show that complexity-aware planning improves several aspects of long-horizon generation, though not every metric changes by the same margin. On Tora, generation-time planning reduces fluid trajectory error from 122.61 to 102.20 and increases aesthetic quality, imaging quality, and physical commonsense. Object trajectory error, however, increases because planning alone cannot force the denoising trajectory to follow object support once the backbone drifts. On VACE, planning improves object trajectory error, subject consistency, aesthetic quality, and imaging quality, but does not improve fluid trajectory error.

Table 3. **Effectiveness of physics-induced latent control on multiple backbones.**

Method	Size	Obj. Traj. Err. ↓	Flu. Traj. Err. ↓	Subj. Cons. ↑	Temp. Flick. ↑	Aesth. Qual. ↑	Img. Qual. ↑	Physical Commonsense ↑
WanMove	14B	89.62	70.54	0.884	0.959	0.536	0.658	0.188
Tora	5B	54.68	80.77	0.907	0.975	0.508	0.642	0.223
Tora w/ Latent Control	5B	**31.42**	**75.77**	**0.964**	**0.993**	**0.527**	**0.661**	**0.226**
VACE	1.3B	**36.34**	81.23	0.778	0.969	0.512	0.599	**0.144**
VACE w/ Latent Control	1.3B	39.32	**74.79**	**0.837**	**0.985**	**0.534**	**0.642**	0.134

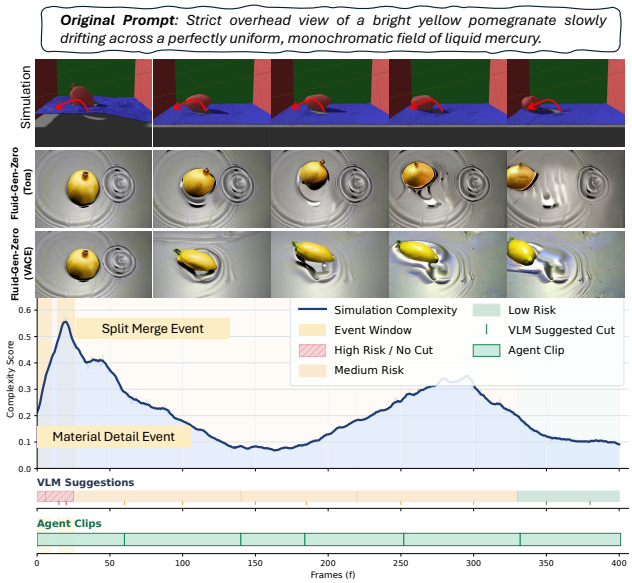

Figure 7. **Generation-time planning and controlled generation.** The top rows show generated videos from Fluid-Gen-Zero on different backbones, and the bottom panels visualize the simulation-derived complexity curve.

4.4. Effectiveness of Physics-Induced Latent Control

We evaluate physics-induced latent control under each backbone's shorter native generation setting. Unlike planning-only generation, latent control directly intervenes in the frozen denoising trajectory by injecting simulation-derived reference latents with object/fluid-specific schedules. As shown in Table 3, this produces strong gains on Tora, reducing object and fluid trajectory errors from 54.68/80.77 to 31.42/75.77 while improving all reported quality metrics. On VACE, latent control improves fluid alignment and all VBench dimensions, but slightly increases object trajectory error, reflecting a backbone-dependent trade-off: VACE already preserves object layout well in this setting, while additional fluid wrapping prioritizes liquid motion and visual consistency.

4.5. User Study

We conduct a two-alternative forced-choice (2AFC) user study to evaluate human preferences. As shown in Table 4, human raters strongly prefer Ours (VACE) over VACE, with 84.29% preference for simulation alignment, 87.14% for temporal consistency, and 87.14% for overall quality. Ours (VACE) is also preferred over WanMove on all three axes. Since WanMove uses a much larger 14B backbone, this result suggests that users perceive benefits from simulation-grounded control even when our method is built on a smaller generator. Tora-based variant also receives consistent but more moderate preference over both Tora and WanMove. Detailed user studies are provided in Appendix G.

Table 4. **User preference study.** Participants select the preferred video along three axes. Values greater than 50% imply our method is preferred.

Comparison	Sim. Align. ↑	Temp. Cons. ↑	Overall Pref. ↑
Ours (Tora) vs. Tora	61.43%	60.00%	62.86%
Ours (VACE) vs. VACE	84.29%	87.14%	87.14%
Ours (Tora) vs. WanMove	62.86%	55.71%	52.86%
Ours (VACE) vs. WanMove	67.14%	60.00%	54.29%

5. Conclusion

We presented **Fluid-Gen-Zero**, a training-free framework that gracefully decouples physical reasoning from appearance synthesis. By leveraging simulation as a structural motion prior for frozen video generators, our approach introduces two portable mechanisms: complexity-aware temporal planning to organize generation clips and region-aware latent wrapping to guide the denoising trajectory. Experiments across multiple backbones demonstrate significant gains in simulation alignment, temporal stability, and human preference without parameter updates. These results establish physics awareness as a plug-and-play, inference-time capability for the next generation of video foundation models. Our **limitations** are twofold: First, physical guidance quality depends on reconstruction and simulation accuracy; errors in perception or parameterization can yield context-misaligned rollouts (see Appendix F.3). Second, relying on coarse intermediate conditions, e.g., projected masks and optical flow, underspecifies fine-grained phenomena like fluid-solid coupling, leaving such details to the generative prior. Additionally, long-horizon sequences may still accumulate appearance or camera drift due to error propagation across successive clips.

References

Agarwal, N., Ali, A., Bala, M., Balaji, Y., Barker, E., Cai, T., Chattopadhyay, P., Chen, Y., Cui, Y., Ding, Y., et al. Cosmos world foundation model platform for physical ai. *arXiv preprint arXiv:2501.03575*, 2025.

Alimama Creative Team. FLUX-Controlnet-Inpainting. https://github.com/alimama-creative/FLUX-Controlnet-Inpainting, 2024. GitHub repository.

Authors, G. Genesis: A generative and universal physics engine for robotics and beyond, December 2024. URL https://github.com/Genesis-Embodied-AI/Genesis.

Bansal, H., Lin, Z., Xie, T., Zong, Z., Yarom, M., Bitton, Y., Jiang, C., Sun, Y., Chang, K.-W., and Grover, A. Videophy: Evaluating physical commonsense for video generation. *arXiv preprint arXiv:2406.03520*, 2024.

Bowen Wen, Wei Yang, J. K. S. B. FoundationPose: Unified 6d pose estimation and tracking of novel objects. In *CVPR*, 2024.

Bridson, R. *Fluid simulation for computer graphics*. AK Peters/CRC Press, 2015.

Brooks, T., Peebles, B., Holmes, C., DePue, W., Guo, Y., Jing, L., Schnurr, D., Taylor, J., Luhman, T., Luhman, E., et al. Video generation models as world simulators. *OpenAI Blog*, 2024.

Cai, Y., Li, K., Jia, M., Wang, J., Sun, J., Liang, F., Chen, W., Juefei-Xu, F., Wang, C., Thabet, A., et al. Phygdpo: Physics-aware groupwise direct preference optimization for physically consistent text-to-video generation. *arXiv preprint arXiv:2512.24551*, 2025.

Chen, B., Jiang, H., Liu, S., Gupta, S., Li, Y., Zhao, H., and Wang, S. Physgen3d: Crafting a miniature interactive world from a single image. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 6178–6189, 2025a.

Chen, H. H., Huang, H., Chen, Q., Yang, H., and Lim, S.-N. Hierarchical fine-grained preference optimization for physically plausible video generation. In *NeurIPS*, 2025b.

Chu, R., He, Y., Chen, Z., Zhang, S., Xu, X., Xia, B., Wang, D., Yi, H., Liu, X., Zhao, H., et al. Wan-move: Motion-controllable video generation via latent trajectory guidance. *arXiv preprint arXiv:2512.08765*, 2025.

DeepMind, G. Veo-3: A scalable and controllable text-to-video model. https://storage.googleapis.com/deepmind-media/veo/Veo-3-Tech-Report.pdf, 2025. Technical Report, Google DeepMind.

Foo, L. G., Huang, M. H., Lattas, A., Moschoglou, S., Beeler, T., and Theobalt, C. Physical simulator in-the-loop video generation. *arXiv preprint arXiv:2603.06408*, 2026.

Hu, Y., Fang, Y., Ge, Z., Qu, Z., Zhu, Y., Pradhana, A., and Jiang, C. A moving least squares material point method with displacement discontinuity and two-way rigid body coupling. *ACM Transactions on Graphics (TOG)*, 37(4): 1–14, 2018.

Hu, Y., Li, T.-M., Anderson, L., Ragan-Kelley, J., and Durand, F. Taichi: a language for high-performance computation on spatially sparse data structures. *ACM Transactions on Graphics (TOG)*, 38(6):201, 2019.

Huang, Z., He, Y., Yu, J., Zhang, F., Si, C., Jiang, Y., Zhang, Y., Wu, T., Jin, Q., Chanpaisit, N., Wang, Y., Chen, X., Wang, L., Lin, D., Qiao, Y., and Liu, Z. VBench: Comprehensive benchmark suite for video generative models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2024.

Jiang, Z., Han, Z., Mao, C., Zhang, J., Pan, Y., and Liu, Y. Vace: All-in-one video creation and editing. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 17191–17202, 2025.

Ke, B., Qu, K., Wang, T., Metzger, N., Huang, S., Li, B., Obukhov, A., and Schindler, K. Marigold: Affordable adaptation of diffusion-based image generators for image analysis, 2025.

Kuaishou. Kling. https://kling.kuaishou.com/, 2024. Accessed: 2025-05-04.

Li, C., Michel, O., Pan, X., Liu, S., Roberts, M., and Xie, S. Pisa experiments: Exploring physics post-training for video diffusion models by watching stuff drop. *arXiv preprint arXiv:2503.09595*, 2025a.

Li, Q., Xing, Z., Wang, R., Zhang, H., Dai, Q., and Wu, Z. Magicmotion: Controllable video generation with dense-to-sparse trajectory guidance. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 12112–12123, October 2025b.

Li, Y., Wang, X., Zhang, Z., Wang, Z., Yuan, Z., Xie, L., Shan, Y., and Zou, Y. Image conductor: Precision control for interactive video synthesis. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 39, pp. 5031–5038, 2025c.

Li, Z., Yu, H.-X., Liu, W., Yang, Y., Herrmann, C., Wetzstein, G., and Wu, J. Wonderplay: Dynamic 3d scene generation from a single image and actions. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 9080–9090, 2025d.

Lin, H., Chen, S., Liew, J. H., Chen, D. Y., Li, Z., Shi, G., Feng, J., and Kang, B. Depth anything 3: Recovering the visual space from any views. *arXiv preprint arXiv:2511.10647*, 2025.

Ling, P., Bu, J., Zhang, P., Dong, X., Zang, Y., Wu, T., Chen, H., Wang, J., and Jin, Y. Motionclone: Training-free motion cloning for controllable video generation. *arXiv preprint arXiv:2406.05338*, 2024.

Liu, S., Zeng, Z., Ren, T., Li, F., Zhang, H., Yang, J., Li, C., Yang, J., Su, H., Zhu, J., et al. Grounding dino: Marrying dino with grounded pre-training for open-set object detection. *arXiv preprint arXiv:2303.05499*, 2023.

Liu, W., Chen, Z., Li, Z., Wang, Y., Yu, H.-X., and Wu, J. Realwonder: Real-time physical action-conditioned video generation. *arXiv preprint arXiv:2603.05449*, 2026.

Macklin, M. Warp: A High-performance Python Framework for GPU Simulation and Graphics, March 2022. URL https://github.com/NVIDIA/warp. NVIDIA GPU Technology Conference (GTC).

Meng, C., He, Y., Song, Y., Song, J., Wu, J., Zhu, J.-Y., and Ermon, S. Sdedit: Guided image synthesis and editing with stochastic differential equations. *arXiv preprint arXiv:2108.01073*, 2021.

Meng, F., Liao, J., Tan, X., Shao, W., Lu, Q., Zhang, K., Cheng, Y., Li, D., Qiao, Y., and Luo, P. Towards world simulator: Crafting physical commonsense-based benchmark for video generation. *arXiv preprint arXiv:2410.05363*, 2024.

Monaghan, J. J. Smoothed particle hydrodynamics. *In: Annual review of astronomy and astrophysics. Vol. 30 (A93-25826 09-90), p. 543-574.*, 30:543–574, 1992.

Namekata, K., Bahmani, S., Wu, Z., Kant, Y., Gilitschenski, I., and Lindell, D. B. Sg-i2v: Self-guided trajectory control in image-to-video generation. *arXiv preprint arXiv:2411.04989*, 2024.

Ravi, N., Gabeur, V., Hu, Y.-T., Hu, R., Ryali, C., Ma, T., Khedr, H., Rädle, R., Rolland, C., Gustafson, L., Mintun, E., Pan, J., Alwala, K. V., Carion, N., Wu, C.-Y., Girshick, R., Dollár, P., and Feichtenhofer, C. Sam 2: Segment anything in images and videos. *arXiv preprint arXiv:2408.00714*, 2024. URL https://arxiv.org/abs/2408.00714.

Singer, A., Rotstein, N., Mann, A., Kimmel, R., and Litany, O. Time-to-move: Training-free motion controlled video generation via dual-clock denoising. *arXiv preprint arXiv:2511.08633*, 2025.

Singh, A., Fry, A., Perelman, A., Tart, A., Ganesh, A., El-Kishky, A., McLaughlin, A., Low, A., Ostrow, A., Ananthram, A., et al. Openai gpt-5 system card. *arXiv preprint arXiv:2601.03267*, 2025.

Stam, J. Stable fluids. In *Seminal Graphics Papers: Pushing the Boundaries, Volume 2*, pp. 779–786. 2023.

Stomakhin, A., Schroeder, C., Chai, L., Teran, J., and Selle, A. A material point method for snow simulation. *ACM Transactions on Graphics (TOG)*, 32(4):1–10, 2013.

Sutton, R. S. The bitter lesson. http://www.incompleteideas.net/IncIdeas/BitterLesson.html, March 2019. Accessed: 2026-05-04.

Team, T. H. F. M. Hunyuanvideo 1.5 technical report, 2025. URL https://arxiv.org/abs/2511.18870.

Todorov, E., Erez, T., and Tassa, Y. Mujoco: A physics engine for model-based control. In *2012 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pp. 5026–5033. IEEE, 2012. doi: 10.1109/IROS.2012.6386109.

Wang, A., Huang, H., Fang, Z., Yang, Y., and Ma, C. ATI: Any trajectory instruction for controllable video generation. *arXiv preprint*, arXiv:2505.22944, 2025a.

Wang, J., Ma, A., Cao, K., Zheng, J., Zhang, Z., Feng, J., Liu, S., Ma, Y., Cheng, B., Leng, D., et al. Wisa: World simulator assistant for physics-aware text-to-video generation. *arXiv preprint arXiv:2503.08153*, 2025b.

Wang, Z., Yuan, Z., Wang, X., Li, Y., Chen, T., Xia, M., Luo, P., and Shan, Y. Motionctrl: A unified and flexible motion controller for video generation. In *ACM SIGGRAPH 2024 Conference Papers*, pp. 1–11, 2024.

Weijia Wu, Zhuang Li, Y. G. R. Z. Y. H. D. J. Z. M. Z. S. Y. L. T. G. D. Z. Draganything: Motion control for anything using entity representation, 2024.

Xiang, J., Lv, Z., Xu, S., Deng, Y., Wang, R., Zhang, B., Chen, D., Tong, X., and Yang, J. Structured 3d latents for scalable and versatile 3d generation. *arXiv preprint arXiv:2412.01506*, 2024.

Xie, T., Zhao, Y., Jiang, Y., and Jiang, C. Physanimator: Physics-guided generative cartoon animation. *arXiv preprint arXiv:2501.16550*, 2025.

Xue, Q., Yin, X., Yang, B., and Gao, W. Phyt2v: Llm-guided iterative self-refinement for physics-grounded text-to-video generation. In *Proceedings of the Computer Vision and Pattern Recognition Conference*, pp. 18826–18836, 2025.

Yang, Z., Li, L., Wang, J., Lin, K., Azarnasab, E., Ahmed, F., Liu, Z., Liu, C., Zeng, M., and Wang, L. Mm-react: Prompting chatgpt for multimodal reasoning and action. *arXiv preprint arXiv:2303.11381*, 2023.

Yang, Z., Teng, J., Zheng, W., Ding, M., Huang, S., Xu, J., Yang, Y., Hong, W., Zhang, X., Feng, G., et al. Cogvideox: Text-to-video diffusion models with an expert transformer. In *ICLR*, 2025.

Yao, S., Zhao, J., Yu, D., Du, N., Shafran, I., Narasimhan, K. R., and Cao, Y. React: Synergizing reasoning and acting in language models. In *The eleventh international conference on learning representations*, 2022.

Zhan, J., Li, Z., Yu, H.-X., and Wu, J. Perpetualwonder: Long-horizon action-conditioned 4d scene generation. *arXiv preprint arXiv:2602.04876*, 2026.

Zhang, Z., Liao, J., Li, M., Dai, Z., Qiu, B., Zhu, S., Qin, L., and Wang, W. Tora: Trajectory-oriented diffusion transformer for video generation. In *Proceedings of the Computer Vision and Pattern Recognition Conference*, pp. 2063–2073, 2025.

Zhou, S., Wang, H., Cheng, H., Li, J., Wang, D., Jiang, J., Jin, Y., Huang, J., Mao, S., Liu, S., et al. Physinone: Visual physics learning and reasoning in one suite. *arXiv preprint arXiv:2604.09415*, 2026.

A. Scene Reconstruction and Simulation Details

A.1. Image-to-Simulation Scene Setup

Structured intent. The simulator and the video generator require different forms of information, so Fluid-Gen-Zero first organizes the input image and prompt into a structured intent record. We define $\mathcal{I} = \{\mathcal{A}, \mathcal{D}, \mathcal{E}\}$, where $\mathcal{A}$ stores appearance attributes for generation, $\mathcal{D}$ stores dynamics attributes for simulation, and $\mathcal{E}$ stores physical event tags for planning. This record is not a full scene graph or a learned world model. It is a compact interface that connects prompt parsing, simulation setup, clip planning, and prompt refinement.

The appearance field $\mathcal{A}$ records visual information that should be preserved by the video generator. It includes object appearance, fluid appearance, material description, lighting, background, camera framing, and style. For example, $\mathcal{A}$ may store that the scene contains a glossy ceramic object, transparent water, warm sunset lighting, or a close-up camera view. These fields are mainly used when constructing clip-level prompts, so that local prompt refinements do not change the original visual identity of the scene.

The dynamics field $\mathcal{D}$ records simulator-facing information. It includes fluid type, approximate material parameters, gravity, external actuation, force direction, flow direction, flow speed, object material, object elasticity, and initial motion cues. For example, prompt words such as "water", "oil", or "mud" are mapped to different viscosity and density ranges, while words such as "pour", "splash", "float", or "sink" provide cues for initial velocity, external force, or object-fluid interaction mode. The values in $\mathcal{D}$ are converted into simulator parameters Θ_{sim} for the physical rollout.

The event set $\mathcal{E}$ stores semantic physical events that are important for generation planning. Each event contains an event type, an importance score, and localization cues that link the event to simulation metrics. For example, contact or impact events are localized using object-fluid distance and motion magnitude; splash events are localized using fluid spread, high optical flow, and mask fragmentation; swirl events are localized using curl-like flow patterns; and occlusion events are localized using mask overlap and silhouette change. These events are later used to compute the semantic complexity term and to build no-cut intervals, so that important physical events are not split across clip boundaries.

The final intent record combines a rule-based record $\mathcal{I}^{\text{rule}}$ and a VLM-refined record $\mathcal{I}^{\text{vlm}}$. The rule-based parser maps common material and motion words to valid parameter ranges, and its outputs dominate simulator-ready numeric fields such as viscosity, density, gravity, force magnitude, and flow speed. The VLM is used to refine visually ambiguous fields, including object identity, visible fluid region, material type, scene context, and event salience. This design keeps physical parameters within valid ranges while still using the VLM for image-grounded semantic interpretation.

Scene decomposition. Given this intent, Fluid-Gen-Zero constructs a compact simulator-ready scene state from the image-prompt pair. The input image is treated as the first video frame and decomposed into object, fluid, and static background regions. This decomposition yields two dynamic particle sets, X_0^o for the object and X_0^f for the fluid, together with a static background reference. Separating these components is important because they obey different modeling assumptions. The object should preserve identity, silhouette, and contact geometry, whereas the fluid should deform, spread, and fragment under gravity and interaction.

Region extraction. We obtain the dynamic regions with open-vocabulary detection and segmentation. Grounding-DINO (Liu et al., 2023) proposes text-grounded boxes, and SAM2 (Ravi et al., 2024) converts them into pixel masks. The fluid support is queried with liquid-related prompts such as water, fluid, and liquid, while the main foreground object is queried separately. When the two masks overlap, object support is removed from the fluid mask so that the simulator receives disjoint object and liquid particles. We then estimate metric depth with Depth Anything 3 (Lin et al., 2025), unproject pixels inside the fluid and background masks through the working pinhole camera model, and use Marigold normals (Ke et al., 2025) as local surface-orientation cues for more stable point-cloud construction.

A.2. Reconstruction and Registration

The object branch must convert a segmented 2D crop into a 3D state that is both visually tied to the input image and geometrically usable by the simulator. We use the segmented object crop as the image-conditioned reconstruction input to TRELLIS (Xiang et al., 2024), which produces a textured 3D asset with mesh and Gaussian representations. Because this asset is generated in a canonical coordinate frame, we register it back to the observed RGB-D scene with Foundation-Pose (Bowen Wen, 2024), using the original image, the estimated depth map, the object mask, and the reconstructed mesh. After obtaining the 6-DoF pose, we sample colored points from the posed mesh and refine their translation and isotropic

scale against the depth-lifted object support. The sampled mesh point cloud is centered on the observed object point cloud, and its scale is adjusted by matching robust radial statistics of the two point sets. The registered mesh samples become X_0^o, while the raw depth-lifted object points serve as the observation that anchors pose and scale.

The fluid and background branches are treated more conservatively. Since a single image usually does not determine a complete liquid volume, the fluid support is lifted directly from the metric depth map instead of being inferred as a full 3D asset. The background is reconstructed in the same depth-lifting coordinate system, but it is retained only as static visual context and as geometric evidence for support-plane and camera estimation.

A.3. Simulation Rollout and Generator Conditions

Simulation Rollout. The reconstructed state is then paired with the structured intent to instantiate a bounded simulator configuration. Rather than predicting unconstrained physical constants, Fluid-Gen-Zero maps material words and visual cues to plausible priors for fluid density, viscosity, and object elasticity. Object descriptors such as rigid, metallic, glass, rubber, or soft adjust elastic parameters within plausible ranges. Motion words determine force direction, force magnitude, and flow speed, while scene-level cues determine gravity magnitude and event priorities. The simulator configuration is

$$\Theta_{\text{sim}} = \{\Delta t, N_{\text{sub}}, \mathbf{g}, E^o, \nu^o, \rho^f, \mu^f, u_{0:T_{\text{sim}}-1}\}, \tag{6}$$

where Δt is the simulation time step, N_{sub} is the number of substeps, $\mathbf{g}$ is gravity, (E^o, ν^o) are object material parameters, (ρ^f, μ^f) are fluid material parameters, and $u_{0:T_{\text{sim}}-1}$ denotes optional actuation. Before rollout, the reconstructed point clouds are canonicalized into a simulator coordinate frame. We estimate a local support plane from the reconstructed scene and align gravity with the negative plane normal, so the generated motion respects the apparent scene layout instead of assuming a fixed world-up direction. The fluid is initialized near the support surface as a liquid MPM material point set, and the object is initialized as an elastic material point set. External actuation, when requested by the prompt, is applied along the recovered tangent direction of the support plane.

With this configuration, the physical rollout advances the coupled object-fluid state under gravity, contact, material response, and optional actuation. The object trajectory provides a stable estimate of foreground support, while the fluid trajectory captures coarse liquid transport, spreading, wake formation, and contact-induced motion. The goal is not to reproduce every fine-scale visual phenomenon in the simulator. Instead, the rollout supplies causal motion evidence that is geometrically consistent with the input scene and can be converted into generator-facing conditions. After simulation, the 3D trajectories are projected back to the image plane and resampled to the target generation horizon. The projected particles define time-varying masks $M^o_{1:T}$ and $M^f_{1:T}$, consecutive projected positions define optical-flow or motion-control signals $F_{1:T-1}$, and a lightweight preview $R^{\text{sim}}_{1:T}$ summarizes the physical rollout for VLM observation.

Because the simulator preview is physically informative but not photorealistic, we additionally construct an appearance-preserving simulation reference. Given the simulated foreground mask sequence, we first use the first-frame mask centroid as a SAM2 prompt to recover a precise object mask in the input image. We then remove that object from the first frame using mask-guided inpainting, producing a clean static background. The original object layer is translated according to the centroid displacement of the simulated object masks and alpha-composited over the inpainted background at each frame. This produces the warped reference $V^{\text{sim}}_{1:T}$ and its mask. The reference is not used as the final video. Instead, it provides a latent reference that preserves object appearance, approximate contact timing, and coarse support motion, while leaving texture, lighting, transparency, liquid detail, and high-frequency temporal realism to the pretrained generator.

The resulting simulation bundle $\mathcal{S}$ contains object/fluid trajectories, masks, optical flow, rendered simulation previews, and the appearance-preserving reference. Fluid-Gen-Zero uses this bundle to compute physical, visual, and semantic complexity for generation-time planning, while using the same evidence for latent-space guidance. For each planned clip interval $[s_i, e_i]$, $\mathcal{S}$ is converted into two condition families. The native backbone condition $\mathcal{C}_i$ is produced from the input image, projected flow, and projected masks, and enters the generator through its own conditioning path, such as first-frame anchoring, motion-control videos, source masks, or trajectory-oriented inputs. The simulation reference video and masks for the same interval are instead encoded into the generator latent space and applied through the object and fluid wrapping schedules in Equations (4) and (5). Thus, reconstruction and simulation remain shared across backbones, while only the final preprocessing step and latent-space guidance injection are backbone-specific.

B. Generation-Time Planning Details

B.1. Planning Evidence and VLM Annotations

Given the structured intent $\mathcal{I}$ and simulation bundle $\mathcal{S}$, Fluid-Gen-Zero outputs clip intervals, clip-level prompts, risk summaries, event tags, no-cut intervals, and the simulation conditions used by latent control. The VLM branch adds semantic structure to the metric analysis rather than replacing it. It observes the prompt, input image, sampled simulation previews, optical-flow visualizations, object/fluid masks, and a compact preview of the metric complexity curve. Its output consists of event annotations, risk and no-cut interval suggestions $\mathcal{N}_{\text{vlm}}$, and prompt refinements $\mathcal{P}_{\text{vlm}}$. Each VLM-suggested interval is stored with frame endpoints, a risk label, a no-cut flag, and a confidence score. The rule-based component constrains material parameters, force directions, event categories, and simulator-facing fields to plausible ranges, while the VLM component interprets visual and semantic information that is difficult to express as fixed formulas. This follows the spirit of reasoning-and-acting agents (Singh et al., 2025; Yang et al., 2023; Yao et al., 2022), but the actions in our setting are generation controls such as prompt refinements, risk tags, and protected intervals. We do not let the VLM directly determine the final segmentation. Its annotations can shape semantic emphasis and boundary preferences, but the final clip plan must still satisfy the quantitative simulation complexity and the valid frame counts required by the backbone.

B.2. Complexity Computation and Clip Segmentation

Fluid-Gen-Zero first brings heterogeneous physical, visual, and semantic signals onto a common scale. For a scalar sequence $x = \{x_t\}_{t=0}^{T-2}$, each entry is normalized by

$$\text{norm}(x_t) = \text{clip}\left(\frac{x_t - Q_5(x)}{Q_{95}(x) - Q_5(x) + \epsilon_\nu}, 0, 1\right), \tag{7}$$

where $Q_5(x)$ and $Q_{95}(x)$ are the 5th and 95th percentiles and $\epsilon_\nu > 0$ is a small constant. Percentile normalization prevents a few extreme frames from dominating the entire planning curve.

Physical and visual complexity. After normalization, the physical component aggregates trajectory measurements as $C_t^{\text{phys}} = \sum_{r \in \mathcal{R}_{\text{phys}}} a_r \, \text{norm}(\phi_r(t))$, where $\mathcal{R}_{\text{phys}}$ is the metric set, $\phi_r(t)$ is the raw metric value, and $a_r \geq 0$ is its coefficient. For object or fluid state $q \in \{o, f\}$, we compute the centroid $\mathbf{c}_t^q = (1/N_q) \sum_{n=1}^{N_q} \mathbf{x}_{t,n}^q$. Here N_q is the number of particles in state q, and $\mathbf{x}_{t,n}^q$ is the position of particle n at time t. For each transition, particle displacement is $\|\mathbf{x}_{t+1,n}^q - \mathbf{x}_{t,n}^q\|_2$. The spatial spread of state q is $s_t^q = (1/N_q) \sum_n \|\mathbf{x}_{t,n}^q - \mathbf{c}_t^q\|_2$, and spread change is $|s_{t+1}^q - s_t^q|$. For interaction scenes, we sample object and fluid particles and compute nearest-neighbor fluid-to-object distances; a scene-adaptive threshold δ_c is set from the initial bounding-box size. The physical metrics are summarized in Table 5.

In parallel, the visual component summarizes image-space difficulty as $C_t^{\text{vis}} = \sum_{r \in \mathcal{R}_{\text{vis}}} b_r \, \text{norm}(\psi_r(t))$, where $\mathcal{R}_{\text{vis}}$ is the visual metric set, $\psi_r(t)$ denotes an image-space statistic, and $b_r \geq 0$ is its coefficient. From optical flow $F_t = (u_t, v_t)$, we compute flow magnitude statistics, active-motion support, saliency concentration, discontinuity, curl, and divergence. From object and fluid masks, we compute area change, boundary density, connected-component fragmentation, fragmentation change, and silhouette change. These terms capture thin splash boundaries, swirling motion, disappearing droplets, split/merge events, and other image-space failure modes that are not fully described by trajectory statistics.

Semantic complexity. Semantic complexity connects the prompt-level event structure to the simulation timeline. For each event $e_j \in \mathcal{E}$, the preferred metric set $\mathcal{M}(e_j)$ specifies which physical or visual signals indicate the event. The event is aligned to the peak of the average normalized response over $\mathcal{M}(e_j)$, and Fluid-Gen-Zero marks a short temporal window around that peak as the event interval. Within this interval, we use a smooth bump response $h_j(t)$ scaled by event importance and alignment confidence; outside the interval, the response is zero. The metric sets used for event alignment are summarized in Table 6.

VLM interval suggestions are represented as $\mathcal{N}_{\text{vlm}} = \{(a_\ell, b_\ell, r_\ell, n_\ell, \xi_\ell)\}_{\ell=1}^{L_{\text{vlm}}}$, where a_ℓ, b_ℓ are interval boundaries, r_ℓ is a risk label, n_ℓ marks no-cut intervals, and ξ_ℓ is confidence. Here L_{vlm} is the number of VLM-suggested intervals. We map the risk label to $\zeta(r_\ell) \in [0, 1]$ and define $g_\ell(t) = \xi_\ell \zeta(r_\ell) \mathbb{1}[a_\ell \leq t \leq b_\ell]$. The semantic complexity is $C_t^{\text{sem}} = \max(\max_j h_j(t), \max_\ell g_\ell(t))$.

Fusion and boundary search. The VLM does not directly output the final fused weights. Instead, the intent parser asks the VLM for continuous constraint-priority scores, rather than discrete labels, for physical, visual, semantic, and temporal factors. The raw VLM scores are treated as non-negative relative priorities and their absolute scale is not used. Only the physical,

Table 5. Physical and visual metrics used for complexity computation. All scalar sequences are robustly normalized before being fused into C_t^{phys} or C_t^{vis}.

Metric	Definition
Physical trajectory metrics, computed for both foreground object and fluid states	
Centroid speed	Frame-to-frame centroid displacement $\|\mathbf{c}_{t+1}^q - \mathbf{c}_t^q\|_2$.
Mean particle speed	Mean particle displacement magnitude over particles.
Particle-speed variance	Variance of particle displacement magnitudes over particles.
High-percentile displacement	95th percentile of particle displacement magnitudes.
Particle acceleration	Mean magnitude of the finite difference of particle displacement vectors.
Spread change	Absolute frame-to-frame change of spatial spread s_t^q.
Occupancy-entropy change	Absolute frame-to-frame change of particle occupancy entropy on a coarse spatial grid.
Contact intensity	Fraction of sampled fluid particles whose nearest sampled object particle is within δ_c.
Contact closeness	Mean exponential closeness $\exp(-d/\delta_c)$ over sampled fluid-to-object nearest distances d.
Contact change	Absolute frame-to-frame change of contact intensity.
Optical-flow metrics	
Mean flow magnitude	Mean flow magnitude over image pixels.
Flow-magnitude variance	Variance of flow magnitude over image pixels.
High-percentile flow magnitude	95th percentile of flow magnitude over image pixels.
Active-motion ratio	Fraction of pixels whose flow magnitude is above a small motion threshold.
Motion saliency concentration	Fraction of total flow magnitude contained in the top 10% highest-motion pixels.
Flow discontinuity	Mean spatial gradient magnitude of the two flow channels.
Flow curl	Mean absolute curl $\|\partial_x v - \partial_y u\|$.
Flow divergence	Mean absolute divergence $\|\partial_x u + \partial_y v\|$.
Mask metrics	
Mask-area change	Absolute frame-to-frame change of foreground-pixel ratio.
Boundary density	Fraction of image pixels lying on the eroded-mask boundary.
Fragmentation	Number of connected foreground components under 8-connectivity.
Fragmentation change	Absolute frame-to-frame change of connected-component count.
Silhouette change	XOR-over-union ratio between consecutive masks.

visual, and semantic scores are used for Equation (2); they are first normalized to obtain $\tilde{w}_k = p_k/(\sum_{j\in\{\text{phys,vis,sem}\}} p_j + \epsilon)$, so $\tilde{w}_k \in [0,1]$ and $\sum_k \tilde{w}_k = 1$. We then interpolate the normalized VLM-derived priority $\tilde{w}$ with a fixed default prior $\bar{w} = (0.40, 0.40, 0.20)$ as $w_k \propto \lambda\bar{w}_k + (1-\lambda)\tilde{w}_k$ for $k \in \{\text{phys}, \text{vis}, \text{sem}\}$, where $\lambda \in [0,1]$ controls the strength of the default prior. In our implementation, $\lambda = 0.5$. The final weights are again normalized to sum to one, hence $w_p, w_v, w_s \in [0,1]$. If one signal family is unavailable, its weight is redistributed over the remaining available families.

Temporal planning then converts the fused transition curve into valid generation clips. Here C_t is defined on the transition from frame t to frame $t+1$, while a boundary b is an integer frame index. If a clip starts at frame s and ends at frame b, the next clip also starts from frame b as a rolling first-frame anchor, and the duplicated anchor frame is removed after stitching. The planner proceeds greedily from left to right. At the current start frame s, it first inspects the complexity values within the maximum feasible horizon and chooses a target length $L^\star$: a high peak selects the short target, a medium mean complexity selects the default target, and otherwise the long target is used. The implementation uses $(L_{\min}, L_{\text{def}}, L_{\max}) = (13, 41, 81)$ frames with high/medium thresholds $(0.72, 0.45)$.

Candidate boundaries are obtained by enumerating all backbone-valid clip lengths

$$L \in \mathcal{L}_s = \{L_{\min}, L_{\min}+4, \ldots, \lfloor L_{\max}^s \rfloor_{4n+1}\}, \tag{8}$$

where $L_{\max}^s = \min(L_{\max}, T-s)$ and $\lfloor\cdot\rfloor_{4n+1}$ denotes the largest valid $4n+1$ frame count not exceeding the current horizon. Thus a full VACE search step considers at most 18 candidates, e.g., $13, 17, \ldots, 81$ frames under the default setting. Each candidate length L induces one candidate boundary $b = s + L - 1$. There is no separate iterative pruning within this set; the planner evaluates the same boundary score in Eq. 3 for every candidate and selects the boundary with the lowest score at each greedy planning step. In our implementation, Eq. 3 is instantiated with $\mathcal{N}(b) = [b-2, b+2]$, $L(b) = L$, $\lambda_c = 1$, $\lambda_l = 0.35/L_{\max}$, $\lambda_e = 1$, and $\lambda_f = 10$. Equivalently, the length term is a normalized penalty $0.35|L - L^\star|/L_{\max}$,

Table 6. Event-to-metric mapping used for semantic complexity. Each event is detected from the prompt and aligned to the simulation timeline by the listed physical or visual signals.

Event type	Preferred metrics $\mathcal{M}(e_j)$
impact/contact	contact intensity, high-percentile flow magnitude, silhouette change
splash	high-percentile fluid displacement, flow discontinuity, fragmentation
split/merge	fragmentation, fluid spread change, flow discontinuity
swirl	flow curl, flow-magnitude variance, fluid particle-speed variance
occlusion	boundary density, silhouette change, contact change
material detail	mean flow magnitude, boundary density
default interaction	contact intensity, mean flow magnitude, fluid spread change

while the no-cut term applies a large penalty to forbidden cuts. Here $\bar{C}_{\mathcal{N}(b)}$ measures local boundary instability, $B_{\text{event}}(b)$ gives a bonus when b lies within two frames of a detected event boundary or a VLM-suggested cut frame, and $\mathcal{F}$ contains forbidden cuts. A cut is forbidden if it falls on a high-complexity transition, inside a high-importance semantic event, or inside a VLM-suggested no-cut interval. After selecting b, Fluid-Gen-Zero emits the clip $[s, b]$, sets the next start frame to $s \leftarrow b$, and repeats until the remaining frames fit into one final segment. This greedy search avoids splitting unstable physical events while keeping each generated clip compatible with the valid frame counts required by the backbone.

B.3. Prompt Planning and Stitching

Once clip boundaries are fixed, Fluid-Gen-Zero assigns each clip a prompt that preserves global scene identity while emphasizing the physical events that occur inside the interval. For each selected clip, the planner starts from the original prompt p, keeps the global scene stem, and adds local information from the semantic events that overlap the clip and the dominant risk tags within the clip. VLM-proposed refinements $\mathcal{P}_{\text{vlm}}$ are used only as optional natural-language edits. The resulting prompt p_i keeps subject identity, material, background, lighting, and camera framing, and then appends local physical emphasis. For example, a contact-heavy interval emphasizes boundary consistency and stable contact geometry, a splash-heavy interval emphasizes droplets and thin liquid sheets, and a topology-change interval emphasizes coherent splitting or merging. The final metadata for each clip contains (s_i, e_i), prompt p_i, risk summary, event tags, no-cut constraints, and the conditions passed to latent control. For long-horizon generation, the last generated frame of clip i is reused as the first-frame anchor for clip $i+1$ when the backbone supports image anchoring. After generation, the duplicated anchor frame is removed before stitching. This rolling anchor reduces clip-boundary drift without requiring any backbone fine-tuning.

C. Physics-Induced Latent Control Details

C.1. Latent Mask Projection

For each generated clip, Fluid-Gen-Zero encodes the simulation-reference video as z_i^{sim} and projects the object and fluid masks to the same latent resolution. The mask projection follows the different roles of objects and fluids. The object mask is projected to latent resolution as a hard support mask M_i^o using hard downsampling or thresholded projection, because object control is intended to preserve geometric support and contact layout. The fluid mask is first smoothed in pixel space by a Gaussian operator $\mathcal{G}_{\sigma_m}$ with bandwidth σ_m and then projected to latent resolution as a soft mask M_i^f, which avoids artificial latent discontinuities around deformable liquid boundaries. Both masks are broadcast over latent channels during wrapping. To avoid applying two incompatible controls to the same latent location, the fluid support excludes the object region: $M_i^{f\setminus o} = M_i^f(1 - M_i^o)$.

C.2. Reference Initialization and Control Schedules

Initialization controls how strongly the simulation reference constrains the early denoising trajectory. Starting from pure noise would require later control steps to correct an unconstrained latent path, whereas starting from the clean simulation reference would over-constrain appearance and may transfer proxy-rendering artifacts. Fluid-Gen-Zero therefore starts from a noised simulation latent $r_{i,k}^{\text{sim}} = (1 - \sigma_k) z_i^{\text{sim}} + \sigma_k \epsilon_i$ at the control starting iteration k_0. This choice anchors coarse layout and motion while preserving enough stochasticity for the pretrained generator to synthesize texture, lighting, transparency, and fine detail. Here $\epsilon_i \sim \mathcal{N}(0, I)$ has the same shape as z_i^{sim}, and σ_k is the scheduler noise level at denoising iteration k.

Thus $r_{i,k}^{\text{sim}}$ interpolates between the clean simulation reference and random noise according to the current denoising stage.

Asymmetric schedules. During denoising, object and fluid regions use asymmetric schedules because they have different physical and visual requirements. Object control uses a binary-indicator schedule $\beta_k^o = \mathbb{1}[k_0 \le k < k_o] \in \{0, 1\}$, where k_o is the endpoint of object control. Within the early window $[k_0, k_o)$, $\beta_k^o = 1$ fully replaces the object region in $\tilde{z}_{i,k+1}$ with the noised simulation reference; outside this window, $\beta_k^o = 0$ leaves the generator's native update untouched. This binary anchoring strongly preserves identity, silhouette, and contact geometry while the global structure is being formed, and then releases control so that later denoising iterations can refine surface appearance. Fluid control instead uses a continuous decaying schedule $\beta_k^f = \alpha_0 \max(0, 1 - (k - k_0)/(k_f - k_0))\mathbb{1}[k_0 \le k < k_f] \in [0, \alpha_0]$, where α_0 is the maximum fluid-control strength and k_f is the endpoint of fluid control. Rather than fully replacing the fluid region, β_k^f linearly blends the simulation reference into the generator's own update, with strength decaying from α_0 at k_0 to 0 at k_f. The softer schedule is important because excessive fluid replacement suppresses transparency, specular highlights, droplets, and turbulent details. The decay lets the pretrained generator refine liquid appearance in later denoising iterations, instead of forcing the fluid region to keep matching the simulation reference.

D. Implementation Details

For open-vocabulary segmentation, we use a LangSAM-style composition of Grounding-DINO-base (Liu et al., 2023) and SAM2.1-Hiera-Large (Ravi et al., 2024). Metric depth is estimated with Depth Anything 3 Large (Lin et al., 2025), image-space normals are estimated with Marigold normals (Ke et al., 2025), and object geometry is reconstructed with the TRELLIS image-to-3D large model (Xiang et al., 2024). The generated object mesh is registered to the original RGB-D observation with FoundationPose (Bowen Wen, 2024), followed by the robust scale-and-translation alignment described in Appendix A.

The warped reference video is produced by a mask-guided inpainting-and-compositing module. We use SAM2 to refine the first-frame object mask from the simulation mask centroid, then use a FLUX-based inpainter (Alimama Creative Team, 2024) to synthesize the background behind the removed object. The inpainting prompt is a neutral background prompt, and we use 24 diffusion steps with a fixed seed for reproducibility. The original object layer is then translated according to the simulated foreground-mask displacement and alpha-composited over the inpainted background to obtain the motion-reference video and its mask.

We use Genesis as the particle-based simulator. Object particles are initialized with an elastic MPM material, and fluid particles are initialized with a liquid MPM material. Unless otherwise specified, the simulator uses time step $\Delta t = 10^{-3}$ and 30 substeps. Gravity magnitude defaults to $9.8\,\mathrm{m/s^2}$, while object elasticity, fluid density, fluid viscosity, force direction, and force magnitude are set by the rule-grounded intent parser. The rendered simulator evidence includes object/fluid trajectories, projected masks, optical flow, a simulation preview, and the appearance-preserving warped reference.

The rule parser is local and deterministic. For VLM-based intent refinement, complexity judging, and prompt refinement, we use the OpenAI API model `gpt-5.4-mini-2026-03-17`. The VLM receives the prompt, input image, sampled simulation previews, optical-flow visualizations, masks, and a compact representation of the metric complexity curve. Frames are sampled every five frames for VLM calls by default. No VLM fine-tuning is performed. When API calls are disabled, the system falls back to the rule-based parser and metric-only planner.

We evaluate on WanMove, Tora, and VACE backbones. For VACE, the optical-flow condition and first-frame anchor are converted into the native source-video/source-mask format, while the warped simulation reference and masks are used for latent control. For Tora, the same simulation bundle is routed through its trajectory-oriented conditioning path, and the warped reference and masks provide the clip-level motion scaffold. Fluid-Gen-Zero plans variable-length clips subject to each backbone's valid frame-count constraints. Later clips are anchored by the previous generated clip to reduce boundary drift. All planning and latent-control operations are performed at inference time, and no video-generator weights are updated.

E. Benchmark Size and Evaluation Scope

Our benchmark contains 50 curated fluid-object examples, from which we use 40 held-out samples for the experiments in the main paper. Each example pairs an initial image and caption with physical references, including object and fluid point-cloud trajectories, object and fluid regions, optical flow, a physics preview, and an appearance-preserving motion reference.

Existing motion-control benchmarks mainly evaluate whether generated videos follow visual motion annotations. Mag-

Table 7. **Comparison with related benchmarks and simulation-based data.** Motion-control benchmarks provide visual trajectory annotations, while simulation-based systems expose richer physical signals. Our benchmark is designed for fluid-object interaction evaluation with mask, 3D particle, and optical-flow references.

Benchmark / Data	Statistics			Trajectory annotations		
	Samples	Frames	Categorization	Mask	3D Points	Optical Flow
MagicBench (Li et al., 2025b)	600	49	×	✓	×	×
MoveBench (Chu et al., 2025)	1,018	81	✓	✓	×	×
RealWonder (Liu et al., 2026)	30	–	×	×	✓	✓
PerpetualWonder (Zhan et al., 2026)	10	–	×	×	✓	✓
Ours	**50**	**401 (physics rollout)**	✓	✓	✓	✓

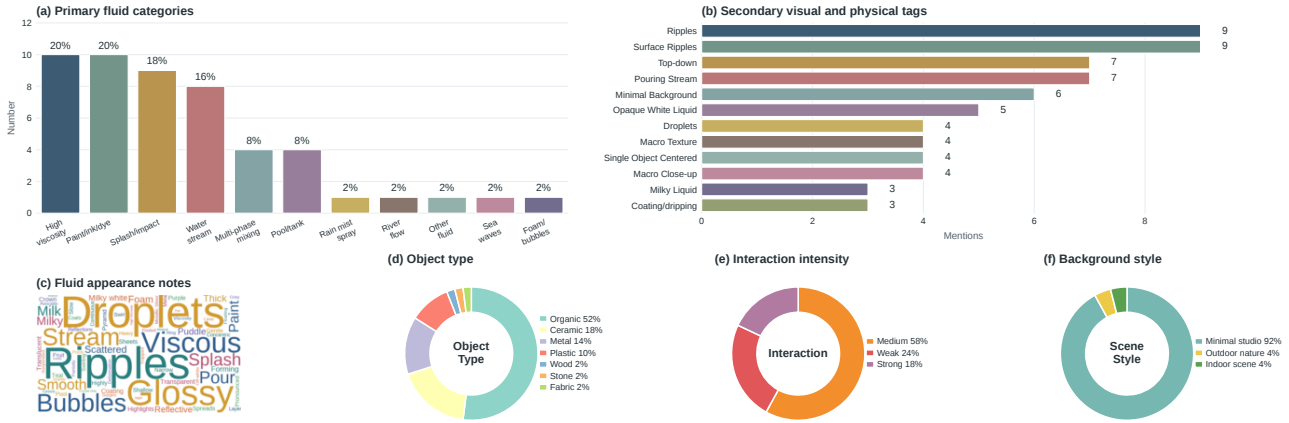

Figure 8. **Benchmark statistics.** Detailed distributions are computed from the diversity annotations of our data.

icBench provides object masks and bounding boxes (Li et al., 2025b), while MoveBench provides category-balanced videos with first-frame masks and object point tracks (Chu et al., 2025). These benchmarks are valuable for visual controllability, but their annotations are primarily in image or video space and do not describe fluid state, material-dependent contact, or coupled object-fluid evolution.

Recent simulation-based generation systems are closer in spirit, but they serve different evaluation goals. RealWonder evaluates 30 action-conditioned samples and uses simulation as an intermediate representation for real-time generation from a single image, supporting forces, robotic actions, camera control, and diverse materials including fluids and granular media (Liu et al., 2026). PerpetualWonder studies closed-loop long-horizon 4D generation on 10 curated scenes, where physical state and visual representation are updated together over multi-step actions (Zhan et al., 2026). These systems demonstrate the value of simulation for generation, while our benchmark focuses specifically on fluid-object interaction evaluation. Every example in our benchmark includes dense object and fluid particle trajectories, projected masks, optical flow, and a 401-frame physical rollout.

Although this size is smaller than standard image or video recognition benchmarks, each sample is substantially more expensive to construct and evaluate than a conventional prompt-video pair. For every example, evaluation requires image understanding, 3D scene recovery, material assignment, object-fluid initialization, physical rollout, projected region and flow references, generation across multiple backbones, and quantitative comparison to the physical reference. The examples are also selected to be difficult rather than redundant. They cover contact, sliding, rotation, impact, submersion, splashing, wake formation, occlusion, viscous deformation, and long-horizon drift, which are precisely the regimes where pretrained video generators tend to fail physically. The 40 held-out samples are evaluated with the same reconstruction, physical rollout, generation-time planning, and latent-space guidance procedure, without per-scene manual retuning. We therefore use the benchmark to evaluate whether the full method remains reliable across diverse hard cases, not to claim exhaustive coverage of every possible fluid, material, camera, or interaction setting. Table 7 summarizes the comparison with related benchmarks and simulation-based systems, and Figure 8 visualizes the internal diversity of our benchmark annotations.

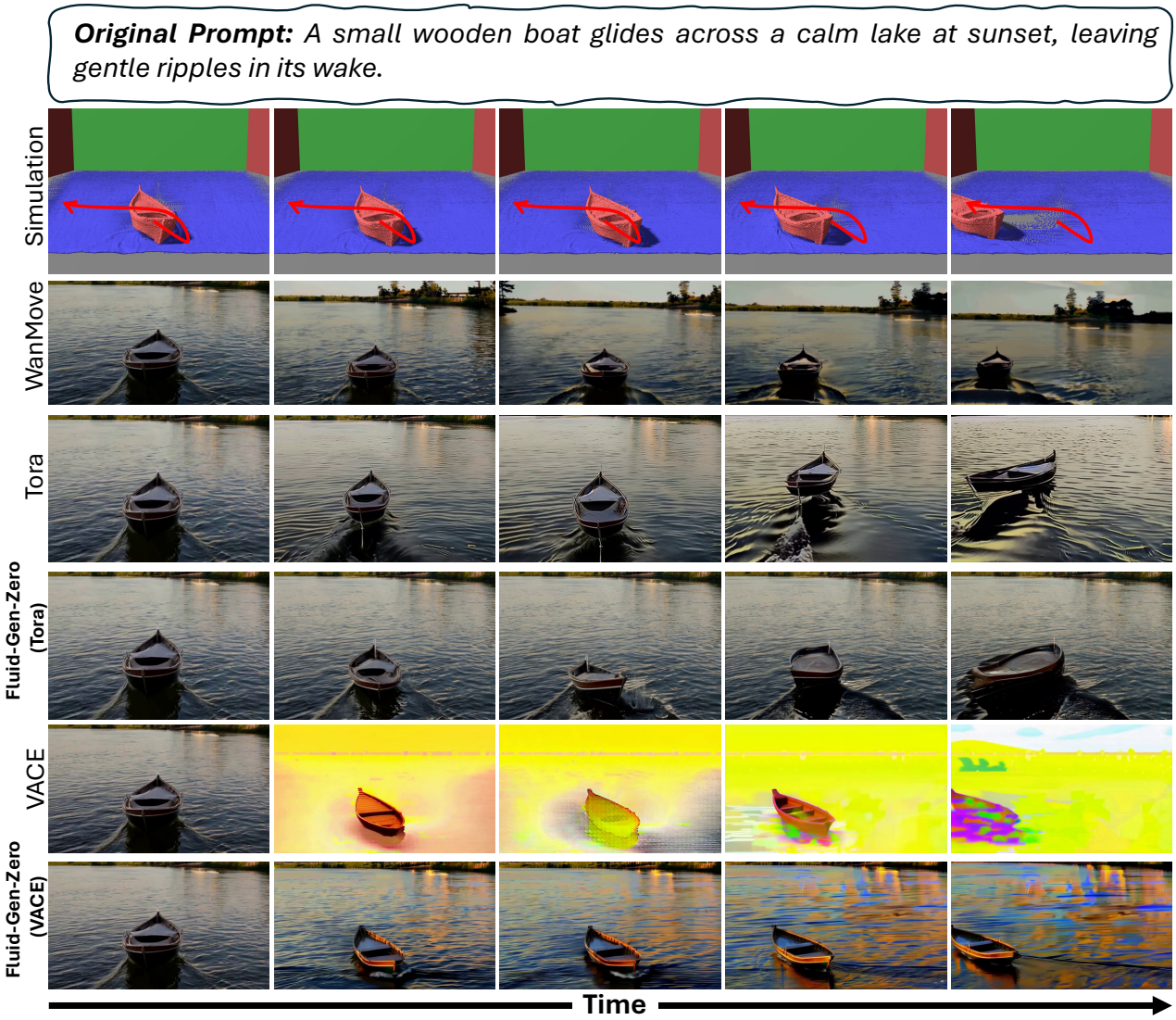

Figure 9. **Additional qualitative comparison results.**

F. Additional Results

F.1. Simulation Study

Simulation enters Fluid-Gen-Zero through two channels, serving as a planning signal and as a latent reference. Table 8 separates these roles. Because some backbones require motion conditions by design, simulation cannot always be removed in a strictly identical way. For VACE, we use the image-text-to-video baseline as the *w/o simulation* setting. For Tora, the native trajectory-control path still requires a motion condition, so the baseline uses simulation-derived conditions but disables both complexity-aware planning and latent-space guidance.

Latent control is responsible for most of the direct trajectory-alignment gain. For Tora, enabling latent control reduces object trajectory error from 107.46 to 56.00 and fluid trajectory error from 122.61 to 97.29. For VACE, the image-text-to-video setting without simulation obtains higher generic perceptual scores, but it has no explicit simulation reference and therefore cannot provide trajectory-level physical control. This highlights a trade-off: unconstrained generation can better exploit the pretrained visual prior, while dense simulation controls, such as VACE optical-flow conditioning, improve physical alignment but may introduce accumulated projection and flow errors over long horizons. Planning contributes more indirectly by improving where and how generation is requested, leading to better fluid alignment or image quality in several settings.

Table 8. **Simulation study.** We evaluate the role of simulation in Fluid-Gen-Zero from two perspectives: simulation as a latent-control signal and simulation as a planning signal.

Method	Obj. Traj. Err. ↓	Flu. Traj. Err. ↓	Subj. Cons. ↑	Temp. Flick. ↑	Aesth. Qual. ↑	Img. Qual. ↑	Physical Commonsense ↑
VACE w/o Sim	–	–	**0.795**	0.962	**0.534**	**0.628**	**0.143**
VACE w/ Latent Control	**37.85**	**123.77**	0.773	0.993	0.428	0.545	0.135
VACE w/ Planning	56.73	134.00	0.718	0.987	0.436	0.540	0.115
Fluid-Gen-Zero (VACE)	43.68	124.00	0.787	**0.993**	0.437	0.590	0.118
Tora w/o Sim	107.46	122.61	0.796	0.980	0.470	0.657	0.123
Tora w/ Latent Control	56.00	97.29	0.901	0.995	0.469	**0.670**	0.176
Tora w/ Planning	121.35	102.20	0.798	0.980	**0.475**	0.662	0.138
Fluid-Gen-Zero (Tora)	**55.92**	**96.78**	**0.901**	**0.995**	0.462	0.668	**0.177**

The full system combines simulation-guided controllability with largely preserved temporal and visual quality.

F.2. Ablation Studies

We also ablate two latent-control design choices on the Tora backbone. **w/o Region-aware** removes the spatial distinction between object and fluid regions and applies a unified control strategy. **w/o Fluid Wrapping** disables the fluid-region latent wrapping, while keeping the remaining simulation conditioning and object-region control unchanged. Table 9 shows

Table 9. **Ablation study of latent control on Tora.** We evaluate key design choices within the latent-control stage.

Setting	Obj. Traj. Err. ↓	Flu. Traj. Err. ↓	Subj. Cons. ↑	Temp. Flick. ↑	Aesth. Qual. ↑	Img. Qual. ↑	Physical Commonsense ↑
Full	**56.00**	97.29	0.901	0.995	**0.469**	0.670	0.176
w/o Region-aware	60.38	95.68	**0.903**	**0.996**	0.468	0.667	**0.184**
w/o Fluid Wrapping	63.06	**92.94**	0.897	0.995	0.464	**0.677**	0.172

that region-aware control is most important for preserving object trajectory. The full Fluid-Gen-Zero design achieves the lowest object trajectory error, indicating that asymmetric object/fluid control is necessary for maintaining object support and contact geometry. Some ablated variants obtain slightly better scores on individual metrics, such as fluid trajectory error, temporal flickering, image quality, or physical commonsense. This does not indicate that the removed design choices are universally preferable. Rather, these metrics expose a trade-off between strict simulation alignment and generative freedom. For example, disabling fluid latent wrapping can allow the backbone to synthesize smoother or more visually natural liquid textures, which may improve image quality or fluid trajectory error, but it weakens object-fluid coupling and increases object trajectory error. Similarly, removing region-aware control applies a less selective constraint, which can occasionally improve global temporal smoothness while reducing the ability to preserve object-specific geometry. We therefore view the full model as the best balanced setting rather than a configuration that must dominate every proxy metric independently.

F.3. Qualitative Results

We provide additional qualitative examples to complement the main paper comparisons. These examples are intended to show how Fluid-Gen-Zero behaves across different scenes. In each sequence, the simulator supplies the coarse physical event order, object trajectory, and fluid support, while the video generator supplies photorealistic appearance. The key comparison is therefore not whether the generated frames exactly reproduce the simulator rendering, but whether they preserve the physically important motion cues while avoiding the appearance artifacts and temporal drift of the baseline generators. Figure 9 highlights a natural water scene, and Figure 10 collects additional object-fluid cases with different materials, camera viewpoints, and interaction types.

Failure Case. Figure 11 shows a representative failure case of Fluid-Gen-Zero. The most common failures occur when the reconstructed simulator state is physically plausible but not sufficiently aligned with the intended image-prompt scene. For example, segmentation leakage, inaccurate depth near transparent fluid, imperfect object reconstruction, or incorrect pose and scale alignment can produce a physical rollout whose object-fluid contact timing differs from the visual prompt. In such cases, Fluid-Gen-Zero may still avoid cutting through high-risk intervals and inject the projected motion reference, but

the generated video can inherit the wrong coarse support, show delayed or excessive fluid response, or produce boundary artifacts around contact regions.

This case illustrates a limitation of our decoupled and training-free design. Fluid-Gen-Zero relies on external perception modules and simulator parameterization to provide a reasonable physical scaffold. The pretrained generator can improve appearance, local realism, and temporal coherence, but it cannot fully correct an upstream physical rollout that is semantically or geometrically mis-specified. Since our framework does not update model parameters or learn to correct reconstruction and simulation errors through gradient-based adaptation, improving robust reconstruction, material estimation, and uncertainty-aware simulation selection remains an important direction for future work.

G. User Study Details

Protocol. We conduct a two-alternative forced-choice (2AFC) user study on 7 complete samples. For each sample, participants are shown the text prompt, input image, simulation reference video, and two anonymized generated videos. Method names are hidden and the order of Video A/B is randomized. Each participant completes 28 pairwise trials, corresponding to 7 samples and 4 comparisons per sample. We collect responses from 10 participants, resulting in 280 pairwise trials and 840 axis-level judgments. Figure 12 shows the screen used in the study.

Evaluation Instructions. Participants answer three preference questions for each pair. For *simulation alignment*, they are instructed to focus on object trajectory, fluid motion, interaction timing, splash or swirl behavior, and the order of major events. The generated video does not need to match the visual appearance of the simulation reference exactly. For *temporal consistency*, they are instructed to focus on flickering, sudden appearance changes, object identity drift, unnatural jumps, broken motion continuity, and visible stitching artifacts. For *overall quality*, they are instructed to consider visual realism, prompt consistency, temporal coherence, and alignment with the simulation reference.

Figure 10. **Additional Fluid-Gen-Zero results.**

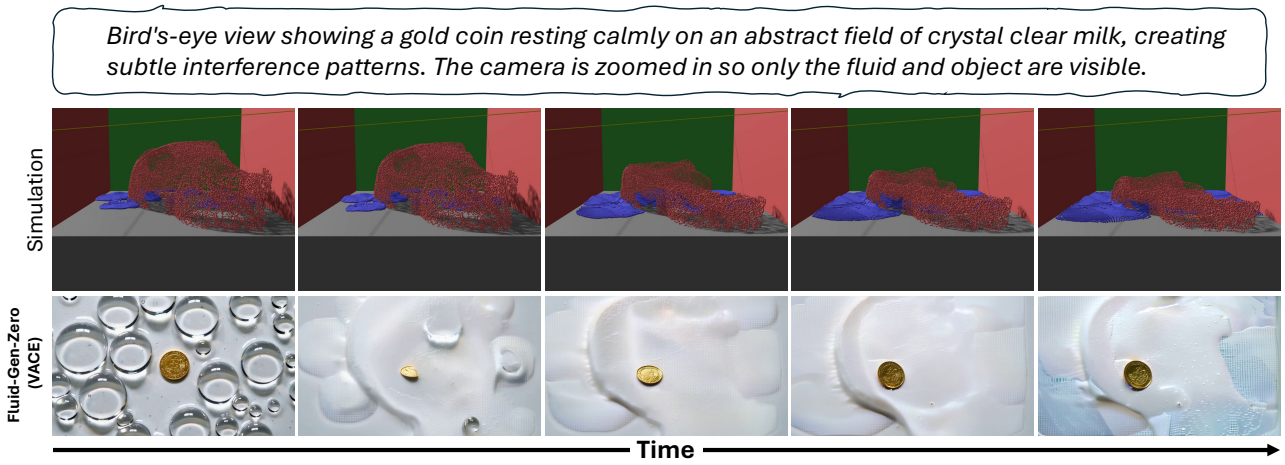

Figure 11. **Failure case.** When reconstruction or simulation is misaligned with the intended scene, Fluid-Gen-Zero may preserve the wrong coarse motion support and produce contact or boundary artifacts.

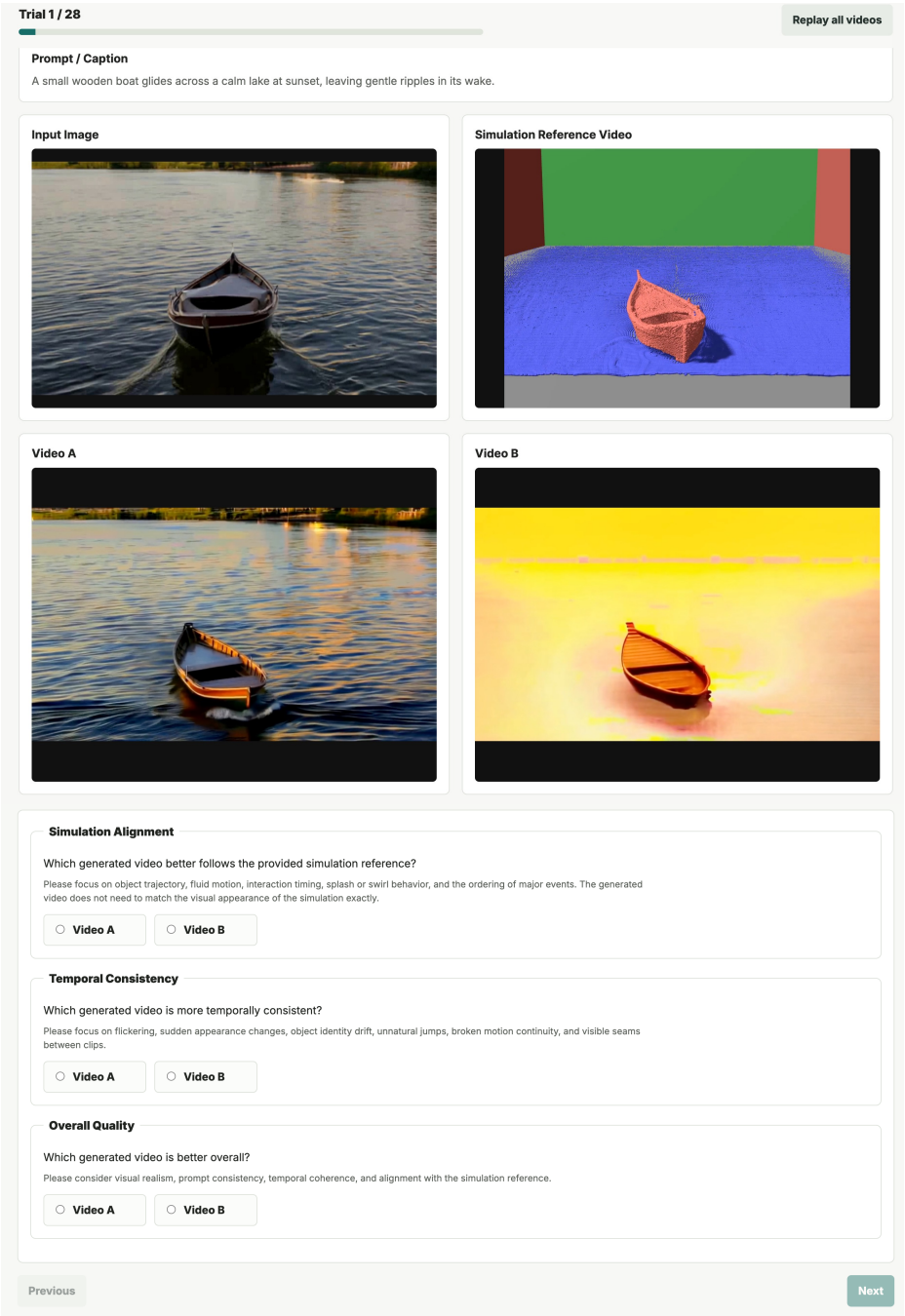

Figure 12. **User study screen.** Participants are shown the prompt, input image, simulation reference, and two anonymized generated videos. They answer three pairwise preference questions for simulation alignment, temporal consistency, and overall quality.