Physical Simulator In-the-Loop Video Generation

Lin Geng Foo[1,5] Mark He Huang[2,3] Alexandros Lattas[4] Stylianos Moschoglou[4]
Thabo Beeler[4] Christian Theobalt[1,5]

[1]Max Planck Institute for Informatics, Saarland Informatics Campus [2]Singapore University of Technology and Design
[3]A*STAR [4]Google [5]Saarbrücken Research Center for Visual Computing, Interaction and Artificial Intelligence

Abstract

Recent advances in diffusion-based video generation have achieved remarkable visual realism but still struggle to obey basic physical laws such as gravity, inertia, and collision. Generated objects often move inconsistently across frames, exhibit implausible dynamics, or violate physical constraints, limiting the realism and reliability of AI-generated videos. We address this gap by introducing Physical Simulator In-the-loop Video Generation (PSIVG), a novel framework that integrates a physical simulator into the video diffusion process. Starting from a template video generated by a pre-trained diffusion model, PSIVG reconstructs the 4D scene and foreground object meshes, initializes them within a physical simulator, and generates physically consistent trajectories. These simulated trajectories are then used to guide the video generator toward spatio-temporally physically coherent motion. To further improve texture consistency during object movement, we propose a Test-Time Texture Consistency Optimization (TTCO) technique that adapts text and feature embeddings based on pixel correspondences from the simulator. Comprehensive experiments demonstrate that PSIVG produces videos that better adhere to real-world physics while preserving visual quality and diversity. Project Page: https://vcai.mpi-inf.mpg.de/projects/PSIVG

1. Introduction

The generation of physically consistent videos represents a key frontier at the intersection of computer vision, graphics, and physical simulation. If achieved, physical consistency significantly enhances visual realism in generated videos, making AI-generated content more compelling for various commercial applications such as film production, virtual reality, and gaming. Ensuring adherence to physics also improves reliability in safety-critical domains like robotics and autonomous driving [46], especially in the use of AI-generated videos to train agent models, directly contributing to the agents' ability to make sound decisions in real-world settings. Due to its widespread importance, this research direction has recently attracted much attention [25, 29, 34].

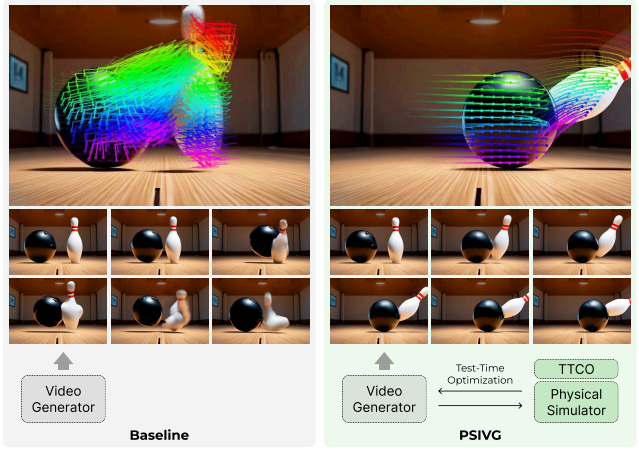

Figure 1. A baseline video generator (left) produces a physically implausible bowling collision with chaotic motion vectors across the video frames. Our PSIVG framework (right) integrates a physical simulator into the generation loop, guiding the video generator to produce a physically plausible and temporally coherent video.

Over recent years, the quality of generated videos [13, 24, 52] has improved tremendously, largely driven by diffusion models and large-scale training. However, even the most advanced video generation models still struggle to capture fundamental physics. Generated scenes often contain objects that lack 3D consistency throughout the frames, vanish abruptly, or move in ways that violate real-world physical laws (see Fig. 1 for an example). Furthermore, crucial physical principles such as gravity, inertia, and collisions are frequently ignored or inaccurately represented. These shortcomings in physical realism have been increasingly observed in recent studies [23, 34], highlighting a critical gap between visual fidelity and physical plausibility.

We observe that a main reason for this is that modern video generation models are often trained on denoising or reconstruction objectives, which mainly encourages the models to "denoise" individual pixels or patches, and thus lack

an explicit understanding of physics since there is no mechanism to enforce physical constraints. To address this, we propose to integrate physical simulators into video diffusion models – a paradigm we refer to as simulation-in-the-loop generation. The simulator serves as a physics-aware constraint, guiding the diffusion model to maintain consistency across time and space. We explore the following research question: How can we effectively incorporate information from the physical simulator into the video diffusion process to achieve physically consistent generation?

In this work, we propose Physical Simulator In-the-loop Video Generation (PSIVG), a novel method for physically consistent video generation from text prompts. Firstly, we generate a *template video* using a pre-trained video generator, which generates the scene background, the camera movements within the scene, the objects, as well as the initial movements which we will use. However, the template video is not physically consistent. To enforce physical consistency, our method relies on a *physical simulator in-the-loop*, which produces physically consistent trajectories for the objects in the video. To incorporate a physical simulator, we first design a perception pipeline (Sec. 3.2.1) that approximately predicts the 3D meshes of foreground objects as well as reconstructs the 4D scene from the template video. Next, this information is used to initialize the scene in the physical simulator (Sec. 3.2.2), including placing and scaling the objects, inferring their physical properties, as well as the initial velocity and rotation of each object. Then, by extracting the RGB, segmentation masks, and pixel-correspondences from the physical simulator, we use them to guide the video generation model to generate physically consistent outputs (Sec. 3.3).

However, we found that directly using the outputs of the physical simulator as conditional inputs for the video generation model [6, 52] is often not enough for high-quality video generation. Specifically, we find that the texture of the objects are often not consistent across the frames, where there is flickering or discoloration of objects during movements and rotations. Texture flickering not only reduces visual quality, but also breaks the perceived temporal coherence necessary for physical realism. To address this, we further design a *test-time texture-consistency optimization* (TTCO) technique to better maintain the textures of moving objects. Intuitively, we optimize learnable parameters such that the generated video more closely follows the pixel-to-pixel correspondences from the physical simulator, improving the texture-consistency during movements and rotations. To achieve a localized optimization targeting the moving foreground objects while maintaining the background, we optimize text embeddings and features corresponding to the foreground objects. TTCO enables the stronger incorporation of physics constraints into the video diffusion model, even when using pre-trained, open-source models that may not be effective at enforcing physical consistency. We highlight that no additional training data is required for our TTCO.

In summary, our contributions are: 1) We propose PSIVG, a novel physical simulator in-the-loop video generation pipeline. PSIVG is the first training-free, inference-time framework to bridge a generative text-to-video pipeline with a 3D physical simulator, enabling on-the-fly physical-consistency guidance for pre-trained video diffusion models. 2) To incorporate the physical simulator in the loop, we design a perception pipeline that reconstructs 3D object meshes and 4D scene motion for initializing the simulator. 3) To further improve the texture consistency of the moving foreground objects, we introduce TTCO, a test-time optimization strategy that improves texture consistency of moving objects guided by simulator correspondences.

2. Related Work

Video Generation Models. Video generation has recently attracted significant attention both in academia and industry [1, 13, 37, 42]. Most recent approaches are based on video diffusion models [17, 18], enabling text-to-video [24, 52] and image-to-video generation [5, 52]. To improve controllability, methods incorporate additional inputs such as masks [2, 53] or multi-modal spatial cues such as masks, depth, and edges [3]. Other works guide motion generation using trajectories [15, 28, 35] or optical flow [6]. Despite rapid progress, achieving physical consistency in generated videos remains difficult, likely due to the inadequacy of reconstruction losses in learning physical principles. We build upon recent diffusion-based generators [6, 52] and introduce physically consistent control signals from a physics simulator. To the best of our knowledge, this is the first approach that integrates a physical simulator in-the-loop into a text-to-video diffusion-based generation pipeline. We further enhance quality through a test-time texture-consistency optimization.

Physically Consistent Generation. Several works have explored physics-aware generative models. Early efforts couple image-based simulators with image generators [29, 33], but these rely on simplified 2D rigid-body assumptions (e.g., spheres, cones), limiting 3D understanding and temporal texture coherence. More recently, a line of works [8, 27, 49] takes input images and user actions (such as manual 2D strokes or rigging) to generate videos. PhysAnimator [49] focuses on animating cartoons, extracting 2D meshes and applying a 2D simulator, then rendering with a fine-tuned sketch-guided diffusion model. PhysGen3D [8] focuses on obtaining a 3D representation for MPM simulation from input images, where forces can then be applied to this simulatable representation to render video sequences from the physical simulator. Differently, we handle open-vocabulary video generation, necessitating our 4D perception pipeline to recover physical states (e.g., object rotations), camera mo-

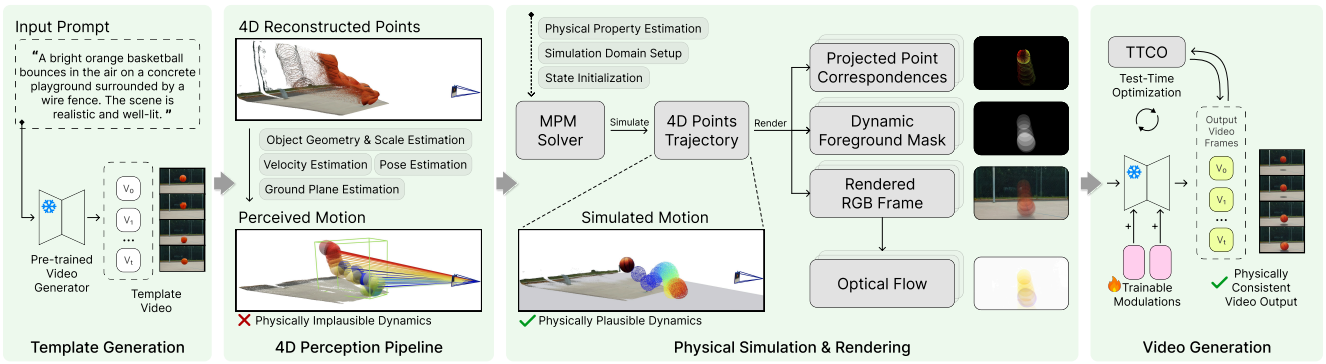

Figure 2. Overview of our Physical Simulator In-the-loop Video Generation (PSIVG) framework. From an input prompt, a template video is first generated, and is processed by our perception pipeline (Sec. 3.2.1). The outputs of the perception pipeline are further processed before being passed into the physical simulator (Sec. 3.2.2). The rendered outputs from the simulator are then used for video generation (Sec. 3.3), and this video generation can be improved with TTCO (see Sec. 3.3.1) for better texture consistency.

tion, and geometry from a generated template video. Our approach of incorporating a pre-trained video diffusion model also provides better video quality and robustness, effectively tolerating reconstruction errors, e.g., errors at the back of reconstructed objects are now refined by the video diffusion model and our TTCO. WonderPlay [27] first generates a 3D Gaussian surfel scene from a single image, then use generated videos to supervise updating of the 3D scene, such that videos can be rendered from the 3D scene. Meanwhile, we perform video refinement directly with our TTCO, which is simpler and more efficient than optimizing a 3D scene; we also support stable 3D object rotations via reconstructed 3D object geometries, which are hard to achieve for a fitted Gaussian surfel scene. For text-to-video generation, some recent studies impose physical consistency via text-driven or LLM-based reasoning: [51] employs physics-grounded prompts, while [31] generates Blender scripts to guide scene construction; such LLM-based explorations are orthogonal to our work. Besides, PISA [25] learns from simulated object interactions and introduces the PisaBench benchmark. Concurrent works also fine-tune diffusion models using physical forces [16] or simulator-derived parameters [45]. In contrast, our approach is training-free and does not require additional data, and instead embeds a physical simulator directly within the text-to-video generation loop, enforcing physically consistent dynamics while preserving the high visual fidelity of diffusion-based synthesis.

Physical Simulators. Physical simulators provide controlled environments for modeling object interactions under physically accurate dynamics. Widely used engines such as PyBullet [10] and MuJoCo [44] support rigid-body dynamics, collision detection, and robotic control, making them popular in reinforcement learning research. Beyond rigid-body simulation, methods based on the Material Point Method (MPM) [41] allow for realistic modeling of deformable materials and have been implemented in frameworks such as Taichi [19] and Warp [32, 54]. Recent efforts also bridge simulation and rendering, for example via 3D Gaussian-based representations [48], and accelerate computation with GPU-optimized engines such as Genesis [4]. Yet, while these simulators are physically accurate, they lack generative capabilities and depend on predefined 3D assets and material properties. Moreover, they often cannot capture complex details such as fine-grained textures, lighting, or fluid dynamics. In contrast, our method couples a physical simulator with a video generative model, combining the physical accuracy of simulation with the visual realism of diffusion models. This enables video generation that is both physically grounded and visually compelling.

3. Method

3.1. PSIVG Pipeline Overview

Our goal is to generate videos whose object motions respect real-world physics while maintaining high visual fidelity. To enable this, we introduce a Physical Simulator In-the-loop Video Generation (PSIVG) framework that integrates physics simulation guidance into a pre-trained video diffusion model. Given an input text prompt, we first generate a *template video* using a pre-trained video generator. This sampled *template video*, although typically problematic in following physical laws, provides essential scene attributes, such as scene composition, camera movements, objects' geometry and textures. Next, we invoke our *perception pipeline* to lift these scene attributes from 2D to 3D and acquire information about its intended 4D dynamics. Using these scene attributes, we initialize a physical simulator and perform forward simulation to obtain physically plausible object trajectories. Finally, the simulator outputs are rendered and fed back into a *video generator*, guiding

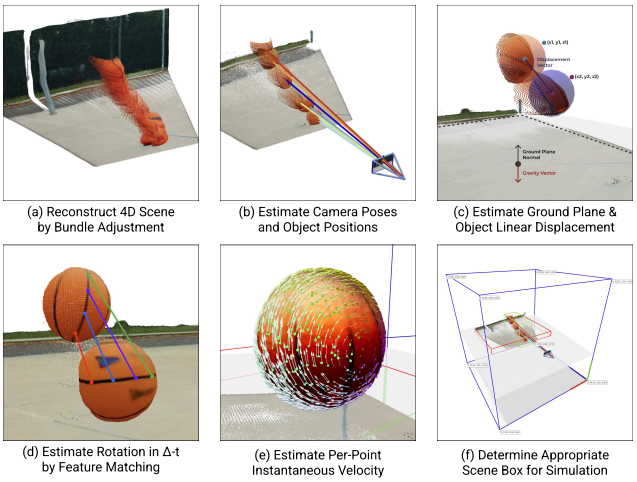

Figure 3. Visualization of the sub-steps in our perception.

it to produce videos that follow a physically consistent motion. To further improve texture consistency, we propose a test-time texture-consistency optimization (TTCO) technique (Sec. 3.3). TTCO enables physics-aware refinement without requiring additional data for retraining the model. Refer to Fig. 2 for an overview.

3.2. Incorporating the Simulator In-the-loop

To incorporate the physical simulator into the generation loop, we do the following after generating the template video: First, we run our *perception pipeline* to obtain 4D scene elements and dynamics from the template video, namely, 3D foreground/background geometries, active object motions, and camera trajectories, which are required for the physical simulation. Next, we run the *physical simulation*, which involves setting up the scene, placing the objects, inferring physical properties, initializing the starting state (e.g., velocity, rotations) and the camera movements. Then, we render the simulated scene and compute physically accurate motion cues as guidance for the video generator, which obey physical principles such as gravity, inertia, and collisions.

3.2.1. Perception Pipeline

The primary objective of the perception pipeline is to transform a generated template video into simulator-ready assets, thereby bridging the gap between the physical simulator and the generative capabilities of video models. To enable physically accurate guidance within the simulator, it is essential to extract three key components from the scene: (1) The dynamics of foreground moving objects; (2) the physical environment with which these objects interact; (3) the camera motion. However, accurately decomposing these components from a generated template video is highly challenging and inherently ill-posed. This difficulty arises because videos produced by vanilla generative models struggle to maintain object permanence and geometric consistency across both spatial and temporal dimensions.

Foreground Object Geometry. To understand the dynamics of moving objects, we need to reconstruct them first. Given an input video, we first detect, ground, and segment all dynamic objects in every frame using off-the-shelf models [9, 30, 38]. For each object instance, we extract an object-centric crop from the first frame (which often has the highest quality), which we feed to InstantMesh [50] for single-image 3D mesh reconstruction. Empirically, leveraging pretrained object priors in image-to-3D models yields more reliable meshes than directly leveraging multi-view methods using different frames in our case, which often suffers inconsistent geometry and texture due to flaws in the video generator.

Background Scene Geometry. Moreover, to recover a spatio-temporally coherent background scene geometry and camera movements from the template video, we perform 4D reconstruction using ViPE [20] which masks the foreground dynamic components from the video and leverages bundle adjustment on key frames (See Fig. 3(a)). We obtain the 3D background geometry by transforming per-frame metric depth pointmaps to a scene-level world frame and aggregate static background points from all frames. From the 4D reconstruction, we also infer the camera poses and rough object positions (see Fig. 3(b)). To enhance the quality and reliability of the reconstructed scene geometry, we apply aggressive sub-sampling and filtering to eliminate floating artifacts and invalid points, which often result from inconsistencies inherent in the template video.

Foreground Object Dynamics. To accurately replicate object dynamics in a physical simulator, it is necessary to determine the object's initial state, including its position, instantaneous velocity. We estimate the initial velocity by decomposing it into linear and rotational components. Specifically, we select two key frames separated by a real-world interval of Δt. The linear velocity is computed as the estimated 3D displacement vector divided by Δt (See Fig. 3(c)). To estimate the rotational velocity, we perform 2D feature matching between object instances in the two frames (See Fig. 3(d)) using SuperGlue [39]. The rotational motion is then isolated by computing a 2D flow field relative to the centroid of the matched feature points (refer to supplemental material for more details). Finally, we combine the estimated linear and rotational components to derive the per-point initial instantaneous velocity for the object (See Fig. 3(e)).

3.2.2. Physical Simulation

We adopt an MPM-based physical simulator [19, 41] to simulate and generate physically accurate scene dynamics and visual guidance for our subsequent video generation. A key step is to initialize the scene (geometries and physical properties) in the simulator such that it recreates the intended dynamics in the template video. Using the outputs from the perception pipeline, the scene initialization includes a few steps, such as determining the simulation domain and

estimating physical properties, which we discuss below.

Simulation Domain. To enable stable simulation and seamless mapping from our perception pipeline to the physical simulator, we first need to determine an appropriate simulation domain that is large enough to include the object and possible range of motion of the object and its interactions with the environment, yet the scene domain should also be as small as possible to improve simulation efficiency. As visualized in Fig. 3(f), we first bound the foreground range of dynamics (illustrated as a green box) and the background geometries (illustrated as a red box), and we use an spatial offset coefficient (C) to determine a cube (illustrated as a blue box) that appropriately houses the 3D scene centered in the middle. Then, we define the domain box as $[0, 2]$ in x, y, z directions and scale, rotate and translate all scene geometries and camera parameters with respect to this simulation domain. By doing so, we can therefore determine the appropriate simulation resolution and metric-to-simulation scale (S) that will also be used to scale the physics constants such as the gravity value and Young's modulus.

Physical Property Estimation. To enable physically plausible interactions, we initialize the physical properties of objects that influence motion and contact dynamics. Specifically, we employ a large vision-language model (GPT-5 [36]) to infer object-specific parameters from the first frame of the template video, guided by a curated text prompt. The model predicts material-related attributes such as density and Young's modulus. However, directly estimating numerical values often yields inconsistent or unreliable results. To address this, we design a hierarchical prompting framework that first queries for intermediate material descriptors, including object composition, elasticity or bounce characteristics, and surface roughness; then, these qualitative properties are mapped to corresponding physical parameters to be used in simulation. The full prompting pipeline and parameter mapping details are provided in the supplementary material.

Simulate and Render. After initializing the scene with the obtained information, we run the forward MPM physical simulation [19, 41], obtaining physically plausible high-resolution particle-level trajectories in 3D space. To obtain explicit guidance signal in pixel space, we render the simulated particle data using Mitsuba [22] into RGB frames, segmentation masks, and frame-to-frame pixel-to-pixel correspondences using the camera poses estimated from the template video. These rendered outputs from the physical simulator are then used to guide the video generation.

Note, that we find that the physical simulator cannot directly replace the generator since the simulator's rendered RGB is often unnatural and unrealistic, due to several reasons: Firstly, the rendered RGB is in a very artificial simulator-like style, and the visual style can be quite unnatural and very different from the style of the intended video background. Furthermore, physical simulators often cannot handle other factors such as lighting and shadows, and the rendering is also often not in high resolution – which all affect the quality of the rendered outputs. Moreover, there are often inaccuracies and imperfections in the 3D object mesh, which leads to unrealistic rendered video from the physical simulator. Therefore, these rendered outputs are often not a good replacement for videos generated by video generation models. Yet, although these renders lack photo-realism, they encapsulate faithful motion physics, which is helpful to guide a video generation model, facilitating the generation of plausible and physically consistent video. We discuss these further in the next section.

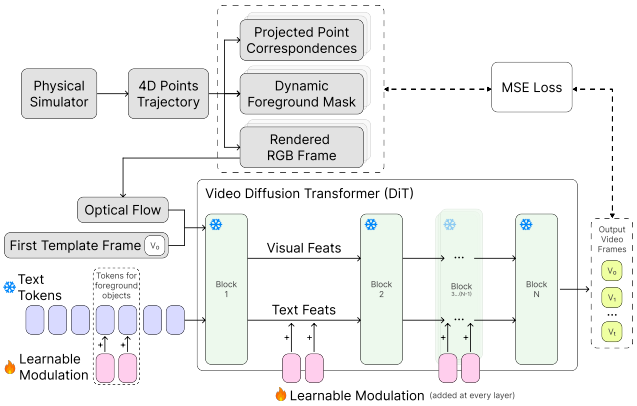

Figure 4. Overview of TTCO. To improve the consistencies of textures, during test time, we add learnable zero-initialized embeddings to the text prompt and features, and optimize them with the outputs from the physical simulator. This allows the generated video to adhere to the simulator trajectories and rotations better, thereby improving the texture consistency.

3.3. Physically-consistent Video Generation

The rendered outputs from the physical simulator, can then be used to guide the video generation. Such guidance can be performed in several ways, including conditioning the video generation on segmentation masks [2] or depths [3]. Here, we use an optical flow-conditioned video generation, model Go-with-the-Flow (GwtF) [6], since optical flow conditioning allows for simultaneously encoding trajectories and rotations, as well as convenient modeling of camera movements with background optical flow.

Specifically, to use GwtF [6], we compute RAFT optical flow [43] from two sources: 1) To get physically consistent foreground motion, we compute the optical flow from the simulator-rendered RGB. 2) To preserve the background scene movements and camera dynamics, we compute optical flow from the template video. These flows are fused with the aid of segmentation masks to form a hybrid flow field, preserving real-world motion cues which are difficult to fit and model in the simulator (e.g., water, foliage) and camera movements, while enforcing physics-based constraints for object motion. Then, the optical flow is used to warp

the noise latents (following GwtF [6]), which is input into the model along with the prompts and the starting frame of the template video. Note, that RAFT optical flow is used instead of the simulator's pixel-to-pixel correspondences, since GwtF is trained with RAFT and attains good performance when RAFT optical flow is used.

3.3.1. Test-time Texture-consistency Optimization

With the above process, we can generate video that follows the general trajectories and movements of the foreground objects. However, even with accurate motion guidance, existing flow-conditioned video generation models [6] still exhibit texture flickering and object appearance drift, often failing to generate objects with consistent textures and colors across the frames. For instance, there may be flickering during certain frames, or the textures of certain objects may change unnaturally during rotations. To address this, we introduce a *test-time texture-consistency optimization* (TTCO) technique to improve the texture-consistency of the generated objects. See Fig. 4 for an overview.

TTCO is a lightweight, test-time procedure that locally adapts the model to maintain texture consistency of foreground objects across frames. During test-time, we optimize the learnable parameters, so that the generated video follows more accurately the trajectory and rotation of the objects in the physical simulation. Specifically, this is done by applying a pixel-correspondence loss using the pixel-to-pixel correspondences from the physical simulator, which encourages the pixel-to-pixel movement between frames to follow the physical simulator's foreground. Note that no additional data are required for TTCO.

Let $\hat{L}_\tau$ denote the latent predicted by the diffusion model at denoising timestep τ. Let $\hat{I}_1$ be the first frame of the template video, and let $W_t(\hat{I}_1)$ denote the warping of the first frame to the t-th frame using simulator pixel correspondences $\{(p_{1,j}, q_{t,j})\}_{j\in J}$, where $p_{1,j}$ and $q_{t,j}$ indicate the j-th corresponding pixel locations between frames 1 and t. The texture consistency loss for the t-th frame is defined as:

$$\mathcal{L}_{\text{tex}}(t) = \sum_{j=1}^{J} \left\| \left[De(h_0(\hat{L}_\tau))\right]_{q_{t,j}} - \left[W_t(\hat{I}_1)\right]_{q_{t,j}} \right\|_2^2, \quad (1)$$

where $h_0(\cdot)$ is the deterministic DDIM-style step mapping to the final denoising iteration [12, 40], and $De(\cdot)$ denotes the decoder. The operator $[\cdot]_q$ retrieves the pixel value at location q. We sum up this loss over all frames: $\mathcal{L}_{\text{TTCO}} = \sum_{t=2}^{T} \mathcal{L}_{\text{tex}}(t)$.

In practice, to implement this, we apply the pixel-to-pixel correspondences from the physical simulator to warp the first frame of the template video, creating a texture-consistent target. Then, we apply a pixel-wise MSE loss between the generated video and the texture-consistent target, and is applied with the aid of the mask such that only the foreground pixels are considered. Note, that because the pixel correspondences are often sparse, i.e. , does not densely cover every pixel of the foreground, we also perform an interpolation operation to compute the dense pixel correspondences of all pixels. Furthermore, in cases with larger object rotations, the objects quickly rotate till none of the pixels were visible in the original first frame, thus the pixel-to-pixel warping loss cannot be directly enforced; in those cases, we use the pixel values of the rendered reconstructed object in the simulator, which can also facilitate texture consistency in the video. During this test-time optimization, we focus on sampling the earlier (i.e., noisier) diffusion steps, which we find is important in guiding the generation of textures.

Our TTCO technique aims to improve the texture consistency of the moving foreground objects, thus we want the fine-tuning to be *localized* to target these foreground objects, while preserving the quality of the background. To achieve this, we optimize only foreground-related parameters. We introduce a learnable residual token added to text embeddings for object phrases, as well as feature-wise modulations in DiT layers corresponding to object tokens. By focusing on the text prompt and tokens of the foreground object, we observe that the impact on the video background is greatly minimized as compared to some alternative techniques, e.g., introducing LoRA layers. This provides localized adaptation, improving object texture stability while often leaving the background untouched, which is a crucial requirement for maintaining visual fidelity. Our observation that text tokens are strongly related to their corresponding foreground objects also aligns with other recent diffusion-based studies [7, 14], further contributing to the growing body of evidence that text-token modulation is a highly effective mechanism for controlling object-specific appearance. Refer to supplementary material for more details.

4. Implementation Details

To generate template videos, we first use SD 3 [11] to generate images with the prompt, and then use CogVideoX-I2V-5B [52] or HunyuanVideo-I2V [24] with these images and prompts to generate template videos. During TTCO, we adopt AdamW optimizer, with LR=2e-4. We run it for 50 iterations with our $\mathcal{L}_{TTCO}$ loss. During TTCO, we also focus on sampling the diffusion steps 700-1000, i.e, the noisier steps, which we find to be important in guiding the generation of textures. Please refer to the supplementary material for more details.

5. Experiments

5.1. Text-to-Video Generation

To evaluate our method, we conduct experiments across diverse text prompts automatically generated by an LLM. The prompts include both single- and multi-object scenes, and videos with either static or dynamic camera motion.

Table 1. Quantitative comparison with existing methods for text-to-video generation.

Type	Method	Motion Controllability		General Video Generation Quality					
		SAM mIoU ↑	Corr. Pixel MSE ↓	CLIP Text ↑	CLIP Img ↑	Subject Consistency ↑	Background Consistency ↑	Motion Smoothness ↑	Temporal Flickering ↑
Text-based	[52] CogVideoX	0.47	0.032	0.34	**0.99**	0.93	0.95	0.98	0.97
	[24] HunyuanVideo	0.46	0.017	**0.35**	**0.99**	**0.95**	**0.96**	**0.99**	0.98
	[25] PISA-Base	0.50	0.012	**0.35**	**0.99**	**0.95**	**0.96**	**0.99**	**0.99**
	[25] PISA-Seg	0.50	0.012	**0.35**	**0.99**	**0.95**	**0.96**	**0.99**	**0.99**
	[25] PISA-Depth	0.51	0.017	**0.35**	**0.99**	0.85	0.92	0.98	0.98
Controllable	[28] MotionClone	0.68	0.019	**0.35**	0.97	0.87	0.92	0.97	0.94
	[35] SG-I2V	0.75	0.021	0.34	0.98	**0.95**	0.95	0.97	0.94
	[47] DragAnything	0.43	0.020	0.34	0.95	0.88	0.92	0.94	0.92
	[26] Image Conductor-Object	0.61	0.022	0.34	0.95	0.84	0.92	0.93	0.90
	[26] Image Conductor-Camera	0.55	0.023	**0.35**	0.95	0.81	0.90	0.92	0.88
	Ours (PSIVG)	**0.84**	**0.007**	**0.35**	**0.99**	**0.95**	**0.96**	**0.99**	0.97

Figure 5. Qualitative comparisons, showing a teddy bear being dropped (left) and objects colliding (right). See Supp. Mat. for more results.

Evaluation Metrics. We evaluate our method from two aspects: motion controllability and video quality. For motion controllability, we measure how well generated objects follow physically-consistent simulated trajectories using SAM-based mask overlap (*SAM mIoU*) and frame-to-frame pixel correspondence error (*Corr. Pixel MSE*). For general quality, we assess text alignment via CLIP similarity (*CLIP Text*); and temporal consistency with CLIP image embedding similarity between consecutive frames (*CLIP Img*), and also using VBench [21] metrics (*subject consistency* and *background consistency*, *motion smoothness*, and *temporal flickering*). See supplementary for more details.

Baselines. First, we compare against open-source T2V models that accept first frame inputs, including CogVideoX [52], HunyuanVideo [24], and variants of PISA [25], which have been trained for better physical accuracy with base finetuning (PISA-Base), segmentation guidance (PISA-Seg), or depth guidance (PISA-Depth). Next, we also compare with several controllable video generation methods, which are based on masks (DragAnything [47]), object trajectories (Image Conductor-Object [26]) or camera trajectories (Image Conductor-Camera [26]), as well as two training-free motion control methods for controlling movements from reference videos (MotionClone [28]) or trajectories (SG-I2V [35]). For these controllable baselines that require additional information, we implement them by applying the *conditioning information from our physical simulator*.

Quantitative Results. Please see Tab. 1 for the results. We observe that our PSIVG consistently achieves the best performance on the motion controllability metrics (SAM mIoU and Corr. Pixel MSE), indicating a strong ability to maintain physically-consistent and coherent motion trajectories across frames as compared to both text-to-video and controllable video generation baselines. Furthermore, although some methods (e.g., PISA-Seg) achieve comparable temporal smoothness and perceptual quality (as seen from motion smoothness and temporal flickering metrics), we find that these models often produce minimal or nearly static movements, where all frames closely resemble the first one, as shown in qualitative results. Consequently, despite appearing temporally stable, such methods exhibit poor motion diversity and fail to capture physically realistic dynamics,

Table 2. User study results against baseline methods.

Method	Preference Rate (%)
CogVideoX	7.2
HunyuanVideo	4.5
PISA-Seg	2.6
SG-I2V	2.5
MotionClone	0.9
Ours (PSIVG)	**82.3**

leading to low scores in the motion accuracy metrics.

Qualitative Results. We provide qualitative comparisons in Fig. 5 across several examples, covering diverse motion scenarios and object interactions. As shown, our method generates videos that exhibit physically consistent and temporally coherent motion. In contrast, existing text-to-video models (e.g., CogVideoX, HunyuanVideo, PISA-Seg) tend to produce visually appealing but physically implausible motion, such as objects floating in midair, fading away, or jumping around. Similarly, controllable video generation approaches (e.g. , MotionClone, SG-I2V) often still struggle with following the trajectory, especially in terms of rotations, and often do not preserve the consistency of the object and background well. Please see the supplemental material for more visualizations.

5.2. User Study

To further assess physical consistency, we conducted a user study involving 32 participants. Each participant was shown sets of videos generated by 5 strong baseline methods and ours, and was asked to select the one that appeared the most physically plausible. As shown in Tab. 2, our method was preferred in 82.3% of the comparisons, substantially outperforming all baseline models. This confirms that human evaluators consistently perceive our generated videos as more physically consistent and natural.

5.3. Ablation Study

Impact of TTCO. Tab. 3 compares results with and without our proposed TTCO. Incorporating TTCO notably improves Corr. Pixel MSE, indicating better alignment with pixel-level motion such as rotations, and slightly boosts SAM mIoU, reflecting more accurate object trajectories. Appearance of the object is also more consistent, as seen from gains in the subject consistency metric. These results demonstrate the effectiveness of TTCO in enhancing texture consistency.

Impact of Prompt-Based Optimization. We ablate the efficacy of our prompt-based optimization design for TTCO, comparing it against a baseline that fine-tunes a LoRA at test-time in Fig. 6. We observe that the LoRA-based design often degrades video quality, particularly in the background, whereas our prompt-based method consistently yields higher quality results. We attribute this improvement to the lightweight, localized nature of prompt-based optimization, which preserves global visual consistency while refining object-level details. Besides, we also tested a design where we directly optimized the object-specific spatio-temporal tokens instead of text tokens, but found that it often produced grid-like artifacts. In comparison, modulating text prompts and tokens is lightweight and works well.

Table 3. Impact of Test-Time Optimization (TTCO).

Setting	SAM mIoU ↑	Corr. Pixel MSE ↓	Subj. Consis. ↑
w/o TTCO	0.82	0.009	0.93
w/ TTCO (ours)	**0.84**	**0.007**	**0.95**

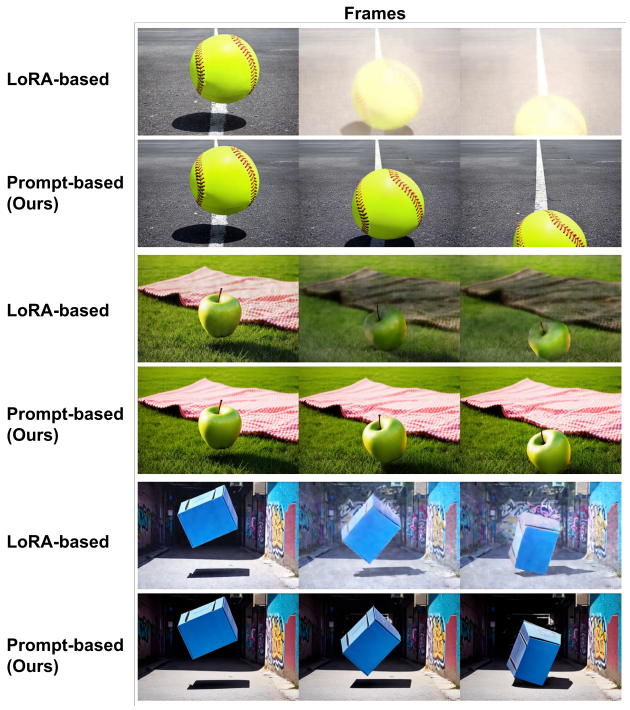

Figure 6. Impact of Prompt-based Optimization Design.

6. Conclusion

We present **PSIVG**, a physical simulator in-the-loop video generation framework that effectively integrates physical simulation into diffusion-based video generation, and enhances texture consistency via TTCO. Extensive experiments demonstrate that PSIVG produces videos with superior physical realism and visual quality compared to existing methods.

Limitations. While effective, our method faces several limitations: (1) Reliance on MPM [41] limits our ability to handle complex agents, such as humans or vehicles, and articulated structures. (2) Limitations in perception quality during initial object reconstruction. (3) We inherit the generative limitations of the GwtF video model [6], e.g., difficulties in generating very small or thin objects. Refer to supplementary material for more details.

Acknowledgments. This work was supported by the Saarbrücken Research Center for Visual Computing, Interaction and Artificial Intelligence (VIA).

References

[1] Meta AI. Meta movie gen. https://ai.meta.com/research/movie-gen/, 2025. Accessed: 2025-10-15. 2

[2] Rick Akkerman, Haiwen Feng, Michael J Black, Dimitrios Tzionas, and Victoria Fernández Abrevaya. Interdyn: Controllable interactive dynamics with video diffusion models. In Proceedings of the Computer Vision and Pattern Recognition Conference, pages 12467–12479, 2025. 2, 5

[3] Hassan Abu Alhaija, Jose Alvarez, Maciej Bala, Tiffany Cai, Tianshi Cao, Liz Cha, Joshua Chen, Mike Chen, Francesco Ferroni, Sanja Fidler, et al. Cosmos-transfer1: Conditional world generation with adaptive multimodal control. arXiv preprint arXiv:2503.14492, 2025. 2, 5

[4] Genesis Authors. Genesis: A universal and generative physics engine for robotics and beyond, 2024. 3

[5] Andreas Blattmann, Tim Dockhorn, Sumith Kulal, Daniel Mendelevitch, Maciej Kilian, Dominik Lorenz, Yam Levi, Zion English, Vikram Voleti, Adam Letts, et al. Stable video diffusion: Scaling latent video diffusion models to large datasets. arXiv preprint arXiv:2311.15127, 2023. 2

[6] Ryan Burgert, Yuancheng Xu, Wenqi Xian, Oliver Pilarski, Pascal Clausen, Mingming He, Li Ma, Yitong Deng, Lingxiao Li, Mohsen Mousavi, et al. Go-with-the-flow: Motion-controllable video diffusion models using real-time warped noise. In Proceedings of the Computer Vision and Pattern Recognition Conference, pages 13–23, 2025. 2, 5, 6, 8

[7] Minghong Cai, Xiaodong Cun, Xiaoyu Li, Wenze Liu, Zhaoyang Zhang, Yong Zhang, Ying Shan, and Xiangyu Yue. Ditctrl: Exploring attention control in multi-modal diffusion transformer for tuning-free multi-prompt longer video generation. In Proceedings of the Computer Vision and Pattern Recognition Conference, pages 7763–7772, 2025. 6

[8] Boyuan Chen, Hanxiao Jiang, Shaowei Liu, Saurabh Gupta, Yunzhu Li, Hao Zhao, and Shenlong Wang. Physgen3d: Crafting a miniature interactive world from a single image. In Proceedings of the Computer Vision and Pattern Recognition Conference, pages 6178–6189, 2025. 2

[9] Ho Kei Cheng and Alexander G Schwing. Xmem: Long-term video object segmentation with an atkinson-shiffrin memory model. In European conference on computer vision, pages 640–658. Springer, 2022. 4

[10] Erwin Coumans and Yunfei Bai. Pybullet, a python module for physics simulation for games, robotics and machine learning. http://pybullet.org, 2016–2019. 3

[11] 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. 6

[12] Lin Geng Foo, Jia Gong, Hossein Rahmani, and Jun Liu. Distribution-aligned diffusion for human mesh recovery. In Proceedings of the IEEE/CVF International Conference on Computer Vision, pages 9221–9232, 2023. 6

[13] Lin Geng Foo, Hossein Rahmani, and Jun Liu. Ai-generated content (aigc) for various data modalities: A survey. ACM Computing Surveys, 57(9):1–66, 2025. 1, 2

[14] Daniel Garibi, Shahar Yadin, Roni Paiss, Omer Tov, Shiran Zada, Ariel Ephrat, Tomer Michaeli, Inbar Mosseri, and Tali Dekel. Tokenverse: Versatile multi-concept personalization in token modulation space. ACM Transactions On Graphics (TOG), 44(4):1–11, 2025. 6

[15] Daniel Geng, Charles Herrmann, Junhwa Hur, Forrester Cole, Serena Zhang, Tobias Pfaff, Tatiana Lopez-Guevara, Yusuf Aytar, Michael Rubinstein, Chen Sun, et al. Motion prompting: Controlling video generation with motion trajectories. In Proceedings of the Computer Vision and Pattern Recognition Conference, pages 1–12, 2025. 2

[16] Nate Gillman, Charles Herrmann, Michael Freeman, Daksh Aggarwal, Evan Luo, Deqing Sun, and Chen Sun. Force prompting: Video generation models can learn and generalize physics-based control signals. In The Thirty-ninth Annual Conference on Neural Information Processing Systems, 2025. 3

[17] Jonathan Ho, Ajay Jain, and Pieter Abbeel. Denoising diffusion probabilistic models. Advances in neural information processing systems, 33:6840–6851, 2020. 2

[18] Jonathan Ho, Tim Salimans, Alexey Gritsenko, William Chan, Mohammad Norouzi, and David J Fleet. Video diffusion models. Advances in neural information processing systems, 35:8633–8646, 2022. 2

[19] Yuanming Hu, Tzu-Mao Li, Luke Anderson, Jonathan Ragan-Kelley, and Frédo Durand. Taichi: a language for high-performance computation on spatially sparse data structures. ACM Transactions on Graphics (TOG), 38(6):1–16, 2019. 3, 4, 5

[20] Jiahui Huang, Qunjie Zhou, Hesam Rabeti, Aleksandr Korovko, Huan Ling, Xuanchi Ren, Tianchang Shen, Jun Gao, Dmitry Slepichev, Chen-Hsuan Lin, Jiawei Ren, Kevin Xie, Joydeep Biswas, Laura Leal-Taixe, and Sanja Fidler. Vipe: Video pose engine for 3d geo-

metric perception. In NVIDIA Research Whitepapers, 2025. 4

[21] Ziqi Huang, Yinan He, Jiashuo Yu, Fan Zhang, Chenyang Si, Yuming Jiang, Yuanhan Zhang, Tianxing Wu, Qingyang Jin, Nattapol Chanpaisit, et al. Vbench: Comprehensive benchmark suite for video generative models. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pages 21807–21818, 2024. 7

[22] Wenzel Jakob, Sébastien Speierer, Nicolas Roussel, and Delio Vicini. Dr. jit: A just-in-time compiler for differentiable rendering. ACM Transactions on Graphics (TOG), 41(4):1–19, 2022. 5

[23] 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 Forty-second International Conference on Machine Learning, 2025. 1

[24] 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. 1, 2, 6, 7

[25] Chenyu Li, Oscar Michel, Xichen Pan, Sainan Liu, Mike Roberts, and Saining Xie. Pisa experiments: Exploring physics post-training for video diffusion models by watching stuff drop. In Forty-second International Conference on Machine Learning, 2025. 1, 3, 7

[26] Yaowei Li, Xintao Wang, Zhaoyang Zhang, Zhouxia Wang, Ziyang Yuan, Liangbin Xie, Ying Shan, and Yuexian Zou. Image conductor: Precision control for interactive video synthesis. In Proceedings of the AAAI Conference on Artificial Intelligence, pages 5031–5038, 2025. 7

[27] Zizhang Li, Hong-Xing Yu, Wei Liu, Yin Yang, Charles Herrmann, Gordon Wetzstein, and Jiajun Wu. Wonderplay: Dynamic 3d scene generation from a single image and actions. In Proceedings of the IEEE/CVF International Conference on Computer Vision, pages 9080–9090, 2025. 2, 3

[28] Pengyang Ling, Jiazi Bu, Pan Zhang, Xiaoyi Dong, Yuhang Zang, Tong Wu, Huaian Chen, Jiaqi Wang, and Yi Jin. Motionclone: Training-free motion cloning for controllable video generation. In The Thirteenth International Conference on Learning Representations, 2025. 2, 7

[29] Shaowei Liu, Zhongzheng Ren, Saurabh Gupta, and Shenlong Wang. Physgen: Rigid-body physics-grounded image-to-video generation. In European Conference on Computer Vision, pages 360–378. Springer, 2024. 1, 2

[30] Shilong Liu, Zhaoyang Zeng, Tianhe Ren, Feng Li, Hao Zhang, Jie Yang, Qing Jiang, Chunyuan Li, Jianwei Yang, Hang Su, et al. Grounding dino: Marrying dino with grounded pre-training for open-set object detection. In European conference on computer vision, pages 38–55. Springer, 2024. 4

[31] Jiaxi Lv, Yi Huang, Mingfu Yan, Jiancheng Huang, Jianzhuang Liu, Yifan Liu, Yafei Wen, Xiaoxin Chen, and Shifeng Chen. Gpt4motion: Scripting physical motions in text-to-video generation via blender-oriented gpt planning. In Proceedings of the IEEE/CVF conference on computer vision and pattern recognition Workshops, 2024. 3

[32] Miles Macklin. Warp: A high-performance python framework for gpu simulation and graphics. https://github.com/nvidia/warp, 2022. NVIDIA GPU Technology Conference (GTC). 3

[33] Antonio Montanaro, Luca Savant Aira, Emanuele Aiello, Diego Valsesia, and Enrico Magli. Motioncraft: Physics-based zero-shot video generation. Advances in Neural Information Processing Systems, 37:123155–123181, 2024. 2

[34] Saman Motamed, Laura Culp, Kevin Swersky, Priyank Jaini, and Robert Geirhos. Do generative video models understand physical principles? arXiv preprint arXiv:2501.09038, 2025. 1

[35] Koichi Namekata, Sherwin Bahmani, Ziyi Wu, Yash Kant, Igor Gilitschenski, and David B. Lindell. SG-i2v: Self-guided trajectory control in image-to-video generation. In The Thirteenth International Conference on Learning Representations, 2025. 2, 7

[36] OpenAI. Gpt-5 is here. https://openai.com/gpt-5/, 2025. Accessed: 2025-10-15. 5

[37] OpenAI. Sora 2 is here. https://openai.com/index/sora-2/, 2025. Accessed: 2025-10-15. 2

[38] Nikhila Ravi, Valentin Gabeur, Yuan-Ting Hu, Ronghang Hu, Chaitanya Ryali, Tengyu Ma, Haitham Khedr, Roman Rädle, Chloe Rolland, Laura Gustafson, Eric Mintun, Junting Pan, Kalyan Vasudev Alwala, Nicolas Carion, Chao-Yuan Wu, Ross Girshick, Piotr Dollar, and Christoph Feichtenhofer. SAM 2: Segment anything in images and videos. In The Thirteenth International Conference on Learning Representations, 2025. 4

[39] Paul-Edouard Sarlin, Daniel DeTone, Tomasz Malisiewicz, and Andrew Rabinovich. Superglue: Learning feature matching with graph neural networks. In Proceedings of the IEEE/CVF conference on computer vision and pattern recognition, pages 4938–4947, 2020. 4

[40] Jiaming Song, Chenlin Meng, and Stefano Ermon. Denoising diffusion implicit models. In International Conference on Learning Representations, 2021. 6

[41] Alexey Stomakhin, Craig Schroeder, Lawrence Chai, Joseph Teran, and Andrew Selle. A material point method for snow simulation. ACM Transactions on Graphics (TOG), 32(4):1–10, 2013. 3, 4, 5, 8

[42] Google AI Studio. Veo 3: Our state-of-the-art video generation model. https://aistudio.google.com/models/veo-3, 2025. Accessed: 2025-10-15. 2

[43] Zachary Teed and Jia Deng. Raft: Recurrent all-pairs field transforms for optical flow. In European conference on computer vision, pages 402–419. Springer, 2020. 5

[44] Emanuel Todorov, Tom Erez, and Yuval Tassa. Mujoco: A physics engine for model-based control. In 2012 IEEE/RSJ international conference on intelligent robots and systems, pages 5026–5033. IEEE, 2012. 3

[45] Chen Wang, Chuhao Chen, Yiming Huang, Zhiyang Dou, Yuan Liu, Jiatao Gu, and Lingjie Liu. Physctrl: Generative physics for controllable and physics-grounded video generation. In The Thirty-ninth Annual Conference on Neural Information Processing Systems, 2025. 3

[46] Yuping Wang, Shuo Xing, Cui Can, Renjie Li, Hongyuan Hua, Kexin Tian, Zhaobin Mo, Xiangbo Gao, Keshu Wu, Sulong Zhou, et al. Generative ai for autonomous driving: Frontiers and opportunities. arXiv preprint arXiv:2505.08854, 2025. 1

[47] Weijia Wu, Zhuang Li, Yuchao Gu, Rui Zhao, Yefei He, David Junhao Zhang, Mike Zheng Shou, Yan Li, Tingting Gao, and Di Zhang. Draganything: Motion control for anything using entity representation. In European Conference on Computer Vision, pages 331–348. Springer, 2024. 7

[48] Tianyi Xie, Zeshun Zong, Yuxing Qiu, Xuan Li, Yutao Feng, Yin Yang, and Chenfanfu Jiang. Physgaussian: Physics-integrated 3d gaussians for generative dynamics. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pages 4389–4398, 2024. 3

[49] Tianyi Xie, Yiwei Zhao, Ying Jiang, and Chenfanfu Jiang. Physanimator: Physics-guided generative cartoon animation. In Proceedings of the Computer Vision and Pattern Recognition Conference, pages 10793–10804, 2025. 2

[50] Jiale Xu, Weihao Cheng, Yiming Gao, Xintao Wang, Shenghua Gao, and Ying Shan. Instantmesh: Efficient 3d mesh generation from a single image with sparse-view large reconstruction models. arXiv preprint arXiv:2404.07191, 2024. 4

[51] Qiyao Xue, Xiangyu Yin, Boyuan Yang, and Wei Gao. Phyt2v: Llm-guided iterative self-refinement for physics-grounded text-to-video generation. In Proceedings of the Computer Vision and Pattern Recognition Conference, pages 18826–18836, 2025. 3

[52] Zhuoyi Yang, Jiayan Teng, Wendi Zheng, Ming Ding, Shiyu Huang, Jiazheng Xu, Yuanming Yang, Wenyi Hong, Xiaohan Zhang, Guanyu Feng, Da Yin, Yuxuan.Zhang, Weihan Wang, Yean Cheng, Bin Xu, Xiaotao Gu, Yuxiao Dong, and Jie Tang. Cogvideox: Text-to-video diffusion models with an expert transformer. In The Thirteenth International Conference on Learning Representations, 2025. 1, 2, 6, 7

[53] Wanyue Zhang, Lin Geng Foo, Thabo Beeler, Rishabh Dabral, and Christian Theobalt. Vhoi: Controllable video generation of human-object interactions from sparse trajectories via motion densification. arXiv preprint arXiv:2512.09646, 2025. 2

[54] Zeshun Zong, Xuan Li, Minchen Li, Maurizio M. Chiaramonte, Wojciech Matusik, Eitan Grinspun, Kevin Carlberg, Chenfanfu Jiang, and Peter Yichen Chen. Neural stress fields for reduced-order elastoplasticity and fracture. In SIGGRAPH Asia 2023 Conference Papers, New York, NY, USA, 2023. Association for Computing Machinery. 3