

PhysCtrl: Generative Physics for Controllable and Physics-Grounded Video Generation

Chen Wang^{1*}, Chuhaoc Chen^{1*}, Yiming Huang¹, Zhiyang Dou²
Yuan Liu³, Jiatao Gu¹, Lingjie Liu¹

¹University of Pennsylvania, ²MIT, ³HKUST * equal contribution
{chenw30, chuhaoc, ymhuang9, jgu32, lingjie.liu}@seas.upenn.edu
frankdou@mit.edu; yuanly@ust.hk

<https://cwchenwang.github.io/physctrl>

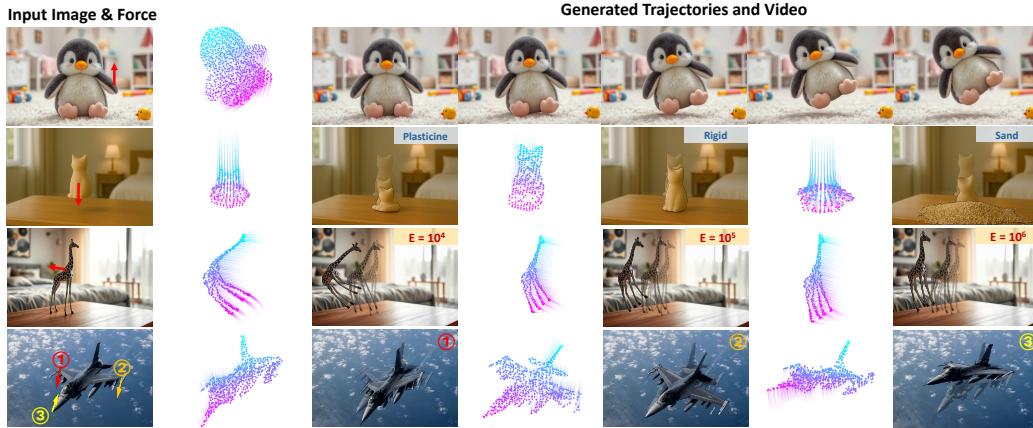


Figure 1: We propose PhysCtrl, a novel framework for physics-grounded image-to-video generation with physical material and force control. PhysCtrl supports generating physics-plausible motion trajectories across multiple materials as control signals (second row), and allows controls over physics parameters (e.g., **Young’s Modulus** E of elastic material (third row)) and **force** (last row). Note that in the bottom three rows, overlaid trajectories and frames use lighter hues for earlier time steps and darker hues for later ones.

Abstract

Existing video generation models excel at producing photo-realistic videos from text or images, but often lack physical plausibility and 3D controllability. To overcome these limitations, we introduce PhysCtrl, a novel framework for physics-grounded image-to-video generation with physical parameters and force control. At its core is a generative physics network that learns the distribution of physical dynamics across four materials (elastic, sand, plasticine, and rigid) via a diffusion model conditioned on physics parameters and applied forces. We represent physical dynamics as 3D point trajectories and train on a large-scale synthetic dataset of 550K animations generated by physics simulators. We enhance the diffusion model with a novel spatiotemporal attention block that emulates particle interactions and incorporates physics-based constraints during training to enforce physical plausibility. Experiments show that PhysCtrl generates realistic, physics-grounded motion trajectories which, when used to drive image-to-video models, yield high-fidelity, controllable videos that outperform existing methods in both visual quality and physical plausibility.

1 Introduction

Video generation has emerged as a transformative technology, powering applications in gaming [7, 75, 14], animation [10, 86, 26], autonomous driving [78, 80], digital avatars [30, 95], robotics [44]. Modern video generative models [57, 86, 88, 4] can produce photo-realistic videos from text or single images. However, they often lack physical plausibility, controllability over dynamic physical behaviors and high fidelity, because they are trained on massive 2D videos in a pure data-driven manner [2, 3].

To achieve physics-grounded video generation, incorporating inductive biases of physical dynamics is crucial. Driven by this, recent works have combined physics simulators [35, 1, 51] with neural representations (e.g., Gaussian splats) to simulate rigid or non-rigid dynamics and render them into videos [85, 36, 46, 8, 71] under scene-specific settings. While physics simulators based on Newtonian mechanics can model the dynamics of diverse real-world systems—including soft/rigid bodies, fluids, and gases [35, 54, 51], they suffer from high computational cost, sensitivity to hyperparameters (e.g., simulation substeps, grid size), numerical instabilities, and trade-offs between generality and accuracy. As a result, when directly using a physics simulator for video generation, people have to tune several hyperparameters and might need to switch simulators with regard to object material (e.g., MPM for elastic and rigid body simulators for rigid). It might also lack robustness and suffer from slow speed (especially for inverse problems).

To address these issues, we propose PhysCtrl, a framework for physics-grounded image-to-video generation with explicit control over physical parameters and external forces. A key component of our framework is a generative physics network, a diffusion-based model that learns the distribution of physical dynamics across various material types. Conditioned on physical parameters and applied forces, it predicts physical dynamics that serve as control signals for pretrained video generative models [24]. In our design, we address two fundamental questions to achieve robust, efficient, and generalizable physics priors for controllable video generation:

1. What is an appropriate representation of physical dynamics for providing control in video models?

We seek a representation that enables efficient control of video models while generalizing across a wide range of materials. Very recent work on controllable video generation [21, 24] has shown that video models can synthesize rich and coherent content from only sparse and explicit point controls. Meanwhile, point clouds offer greater flexibility and generalization for modeling different materials than other explicit representations, such as meshes or voxel grids, making them more suitable for learning-based generative physics networks. Considering these two aspects, we propose to represent physical dynamics as 3D point trajectories, enabling compact motion encoding and seamless integration with video generative models while supporting diverse material types.

2. How to embed generative physics priors across various materials into a network? High-quality and diverse data are essential for learning the distribution of physical dynamics (i.e., *generative physics*). We therefore collect a large-scale synthetic dataset of 550K object animations across four material types (elastic, sand, plasticine, and rigid), capturing complex, physics-grounded dynamics via physics simulators. Using this dataset, we design a diffusion model to generate physics-plausible 3D motion trajectories conditioned on physical conditions. Inspired by particle dynamics [35], where particles interact with neighbors to determine their next state, we introduce a novel spatiotemporal attention block in the diffusion model to emulate these interactions: it first aggregates spatial influences from neighboring points and then predicts each point’s trajectory over time. Finally, to embed explicit physical knowledge directly into the network, we incorporate physics-based constraints during training, ensuring that the generated motions are physics-plausible.

We conduct comprehensive evaluations of our method, demonstrating our model can produce physics-plausible motion trajectories. We further show that the generated trajectories can drive pretrained video models for synthesizing physically plausible image-to-video generation, outperforming existing video generative models in both visual fidelity and physics plausibility. Our key contributions are:

- We introduce PhysCtrl, a novel and scalable framework that represents physics dynamics as 3D point trajectories over time, enabling physics-grounded image-to-video generation with explicit control over physical parameters and external forces.
- We develop a diffusion-based point trajectory generative model equipped with a spatiotemporal attention mechanism and physics-based constraints, efficiently learning generative physical dynamics across four material types.

- We collect a large-scale synthetic dataset of 550K object animations, spanning elastic, sand, plasticine, and rigid materials, using physics simulators. We will release this dataset to support future research in physical dynamics learning.
- We demonstrate the effectiveness of PhysCtrl in generating realistic, physics-grounded dynamics and achieve high-quality image-to-video generation results given user-specified physics parameters and external forces.

2 Related Work

Neural Physical Dynamics Traditionally, physical dynamics are solved with numerical methods such as finite element method (FEM) [96], position-based dynamics (PBD) [55, 50], material point method (MPM) [35], smoothed-particle hydrodynamics (SPH) [17, 58, 40] and mass-spring systems [47]. Physical Informed Neural Networks (PINNs) [59] use neural networks to approximate the solution of partial differential equations and incorporate physics constraints in the loss functions. Combined with neural fields [53], PINNs achieve success in domains like fluids [13, 79] but are limited in per-scene optimization setting. Concurrent work, ElastoGen [19], replaces part of the physics simulation with neural networks for faster inference, but relies on a voxel representation, supports only elastic materials, and requires a full 3D model as input. Graph Neural Networks (GNNs) have emerged as an effective tool for modeling particle interactions with diverse material types [64, 87, 65, 89]. However, such approaches typically rely on next-step predictions for modeling dynamics, making them susceptible to drift and error accumulation over time. In contrast, our method represents objects as flexible point clouds and leverages a spatio-temporal trajectory diffusion model to robustly capture the dynamics of diverse materials in a unified framework.

Controllable Video Generative Models Video generative models are trained on massive text-video paired datasets and achieve high-quality video generation [29, 4, 39, 11, 88]. Existing works have shown that additional control signals can be injected into pretrained models for controllable video generation, such as camera movement [25, 20], human pose [30], and point movement [21, 24, 5]. However, these models lack an understanding of physical laws and thus generate outputs that are often not physically plausible. Furthermore, they cannot support explicit physics control. Our work focuses on generating physics-grounded dynamics that can be used as a physics control signal for video models.

Physics-Grounded Video Generation Existing methods leverage physics simulators to produce physics-grounded videos. One approach reconstructs neural representations from multi-view images, applies simulation on these representations, and then renders the results into video. For example, PhysGaussian [85], Spring-Gaus [93], and Vid2Sim [9] integrate MPM, spring-mass systems, and LBS-based simulation [54] into 3D Gaussians for simulation and rendering. VR-GS [36] applies physics-aware Gaussian Splatting in VR/MR for real-time, intuitive 3D interaction and physics-based editing. PhysDreamer [91] distills motions from video models to estimate physics parameters. These methods are scene-specific and require high-quality 3D reconstruction to achieve good results. Recently, researchers started to combine physics simulators with video generative models. PhysGen [46], PhysGen [8] and PhysMotion [71] generate videos of 2D rigid body dynamics or deformable dynamics. These methods rely on physics simulators to generate dynamics and coarse texture and only use video models for texture refinement. PhysAnimator [84] combines physical simulators and a sketch-guided video diffusion model for animations. Compared with methods that rely on physics simulators, our method embeds physics priors into a diffusion model, which avoids manual hyperparameter tuning and improves numerical stability for dynamics prediction. The predicted dynamics can be used as guidance for video generative models to synthesize physics-grounded and controllable videos. Concurrent works WonderPlay [43] and Force Prompting [22] also investigate using force as the condition signal for video generation.

4D Dynamics Parametric models have been widely used to represent category-specific deformable shapes, such as SMPL and SMAL [49, 97] for human and animal bodies, FLAME [42] for faces, MANO [62] for hands. Recent advances in 4D dynamics have been exploring to capture object dynamics of arbitrary topologies [56, 52, 72, 41, 72, 12] with Neural-ODE and coordinate-MLPs.

With the success of diffusion models [28, 67, 68, 69] on high-quality generation on several modalities, including text [23], image [61, 63], video [29, 27] and 3D [45, 66, 90, 48], researchers have started to learn the distribution of object dynamics with diffusion models [18, 6, 92, 82]. Motion2VecSets [6]

introduced a 4D representation with latent vector sets, and trained a conditional diffusion model for dynamic reconstruction from sparse point cloud sequences. DNF [92] leverages a dictionary-based neural field to learn a compact motion space for unconditional 4D generation. However, these methods are only trained on datasets with a limited number of shapes that contain only human and animal motions, while our method focuses on learning physics-grounded dynamics, which contain a large variety of dynamic phenomena. We also use a more flexible point representation that is better suited for downstream tasks.

3 Preliminary

We generate ground-truth point trajectories for training our generative physics network (also referred to as “physics-grounded trajectory generative model”) on data synthesized by physics simulators, including MPM and rigid body simulators. Here we review the basics of MPM, which form the basis for our physics-aware constraint in Section 4.

Material Point Method Material Point Method (MPM) [70, 60, 38, 35, 33, 31, 85] simulates the deformation of discrete material particles under the assumption of continuum mechanics, where the transformation of each particle from the material space to the world space is defined by a deformation mapping $\mathbf{x} = \phi(\mathbf{X}, t)$, and the associated deformation gradient $\mathbf{F} = \nabla_{\mathbf{X}}\phi(\mathbf{X}, t)$ measures the local deformation of the material such as rotation and stretch. The evolution of ϕ at time t is governed by the conservation of mass and momentum, which can be formulated as

$$\rho \frac{D\mathbf{v}}{Dt} = \nabla \cdot \boldsymbol{\sigma} + \mathbf{f}_{ext} \quad \frac{D\rho}{Dt} + \rho \nabla \cdot \mathbf{v} = 0 \quad (1)$$

where ρ , \mathbf{v} and \mathbf{f}_{ext} denote the density, the velocity field and the per-unit volume external force respectively. The Cauchy stress $\boldsymbol{\sigma} = \frac{1}{\det(\mathbf{F})} \frac{\partial \Psi}{\partial \mathbf{F}}(\mathbf{F}) \mathbf{F}^\top$ and the energy density function $\Psi(\mathbf{F})$ are derived from the deformation gradient \mathbf{F} and physics parameters (e.g. Young’s modulus E and Poisson’s ratio ν) related to specific constitutive models. Based on Equation (1), MPM associates particles with background grids in the simulation, performing a particle-to-grid (P2G) and grid-to-particle (G2P) transfer loop. For stepping t to $t + 1$, the P2G transfer can be formulated as

$$\frac{m_i}{\Delta t} (\mathbf{v}_i^{t+1} - \mathbf{v}_i^t) = - \sum_p V_p^0 \frac{\partial \Psi}{\partial \mathbf{F}}(\mathbf{F}_p^t) \mathbf{F}_p^{t\top} \nabla N_i(\mathbf{x}_p^t) \quad (2)$$

where p and i represent attributes for particle and grid. V_p^0 is the initial particle volume and $N_i(\mathbf{x}_p^t)$ is the B-spline kernel defined on i -th grid evaluated at \mathbf{x}_p^t . Grid mass $m_i^t = \sum_p N_i(\mathbf{x}_p^t) m_p$ and grid momentum $m_i^t \mathbf{v}_i^t = \sum_p N_i(\mathbf{x}_p^t) m_p (\mathbf{v}_p^t + \mathbf{C}_p^t (\mathbf{x}_i - \mathbf{x}_p^t))$ are obtained according to the standard APIC [34], where \mathbf{C}_p^t is the affine matrix. The G2P transfer can be formulated as:

$$\mathbf{C}_p^{t+1} = \frac{4}{(\Delta x)^2} \sum_i N_i(\mathbf{x}_p^t) \mathbf{v}_i^{t+1} (\mathbf{x}_i - \mathbf{x}_p^t)^\top \quad \mathbf{F}_p^{t+1} = (\mathbf{I} + \Delta t \sum_i \mathbf{v}_i^{t+1} \nabla N_i(\mathbf{x}_p^t)^\top) \mathbf{F}_p^t \quad (3)$$

Afterwards, \mathbf{v}_p and \mathbf{x}_p are updated as $\mathbf{v}_p^{t+1} = \sum_i N_i(\mathbf{x}_p^t) \mathbf{v}_i^{t+1}$ and $\mathbf{x}_p^{t+1} = \mathbf{x}_p^t + \Delta t \mathbf{v}_p^{t+1}$.

4 Method

Given a monocular image, our method generates physics-grounded videos with the control signals of physics parameters and external forces. As illustrated in Figure 2, we first lift the input image into 3D points (Section 4.2), and then train a conditional diffusion model to generate physics-grounded point cloud trajectories (Section 4.1) with physics parameters and external forces as conditioning. Finally, we leverage the generated trajectories as condition to pre-trained video models for image-to-video synthesis (Section 4.2).

4.1 Physics-Grounded Generative Dynamics

Our goal is to learn the distribution of physical dynamics across various materials — termed *generative dynamics* — using a diffusion-based model, thereby avoiding the high cost, hyperparameter sensitivity, numerical instabilities, and generality–accuracy trade-offs of classical simulators. We select point clouds as our representation because they flexibly model diverse materials and suffice to control pretrained video models. Specifically, each object is represented by 2048 points in practice; we predict their trajectories over time and use them as control signals for video synthesis.

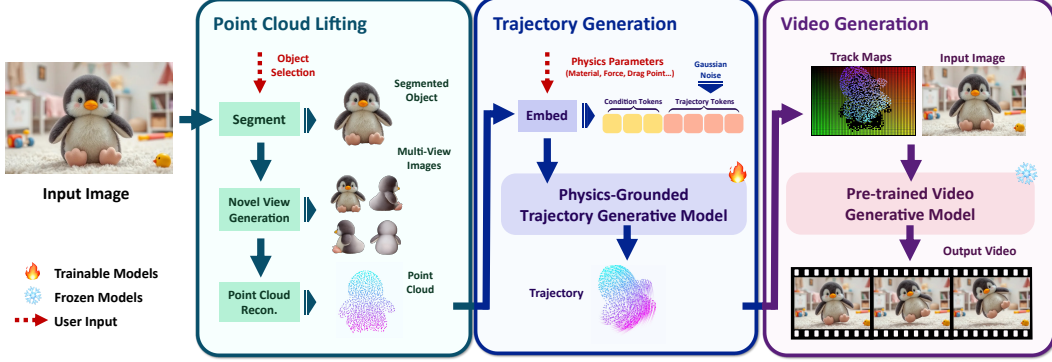


Figure 2: **An overview of PhysCtrl.** Given a single image, we first lift the object in that image into 3D points. We then generate physics-grounded motion trajectories conditioned on physics parameters and external force with a diffusion model, which are then used as strong physics-grounded guidance for image-to-video generation.

4.1.1 Problem Setting

Given an object, represented as a 3D point cloud with N points $\mathbf{P}_0 = \{\mathbf{x}_i^0 \in \mathbb{R}^3\}_{i=1}^N$, and its physics parameters $\{E, \nu\}$, our trajectory generative model generates its dynamics given an initial force. Specifically, the dynamics of the object is represented by the position of each point in future F timesteps $\mathcal{P} = \mathcal{P}^{1:F} = \{\mathbf{P}^f\}_{f=1}^F = \{\{\mathbf{x}_p^f\}_{p=1}^N\}_{f=1}^F$. Denote the force, drag point and boundary condition (floor height) as $\mathbf{f} \in \mathbb{R}^3$, $\mathbf{D} \in \mathbb{R}^3$, and $h \in \mathbb{R}^1$. Thus, the goal of PhysCtrl is to predict \mathcal{P} under the condition $c = \{\mathbf{P}_0, \mathbf{f}, \mathbf{D}, \{E, \nu\}, h, [\text{mat}]\}$. Here, we use an additional [mat] token to denote different materials. In this paper, we cover four different materials: elastic, plasticine, sand, and rigid. Notably, because of our flexible point cloud representation, the model is not limited to these four categories and can be readily extended to other materials, such as fluids, given sufficient computational resources.

We train our trajectory generative model on data from physics simulators—MPM [35] and a rigid-body solver. Simulator hyperparameters (e.g., substeps, grid size) introduce variability that our model, conditioning only on core physics parameters, does not capture directly. To account for this uncertainty, we employ a diffusion model to learn the conditional distribution $p(\mathcal{P}|c)$. Our method can also be extended to learning physics from more simulation methods since it requires only sampled points.

4.1.2 Physics-grounded Trajectory Generative model

Prior trajectory generative models for human motion synthesis [74, 94] typically project all point positions into a single latent space, applying attention to only temporal correlations. This approach is inadequate for our setting (see Figure 1), as it overlooks crucial spatial relationships. While naive 4D attention across both space and time can model spatio-temporal correlations in physics simulation data, it is suboptimal in terms of quality and efficiency due to the combinatorial explosion of spatial points across time steps. Instead, since we aim to model point cloud trajectories with a one-to-one point correspondence across frames, we introduce an efficient attention mechanism tailored for physics simulation data, which first applies spatial attention followed by temporal attention. This design reduces the computational complexity and, more importantly, reflects the underlying process of physics simulation: first integrating information from neighboring points, then propagating forward in time dimension.

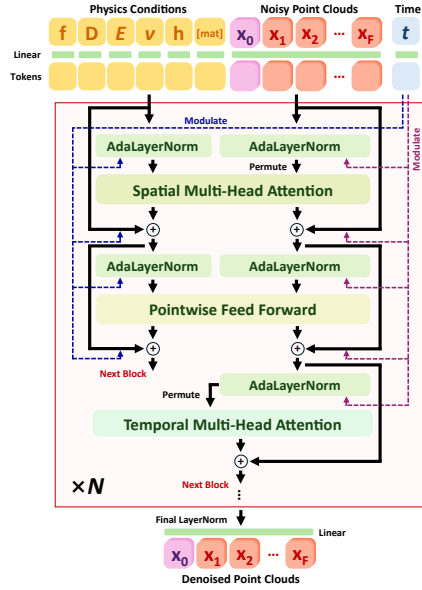


Figure 3: Our trajectory generation architecture which consists of spatial attention and temporal attention in each block.

Specifically, given noisy point cloud sequences, we apply point embedding and project it to latent dimensions, add sinusoidal positional embeddings in both space and time and predict its trajectory offset with our denoising network \mathcal{D} . The core of network \mathcal{D} is a diffusion transformer consisting of a set of spatial-temporal attention blocks as shown in Figure 3. Each block contains two attention layers: spatial attention and temporal attention.

Spatial attention learns the correlation of each point with other points in the same frame with self-attention. To inject physical conditioning c into the attention layer, we first map them into additional tokens using MLPs: $\mathbf{cond} = \text{MLP}_{\text{phys}}([\mathbf{f}; \mathbf{D}; \{E, \nu\}, h, [\text{mat}]] \in \mathbb{R}^{d_c}$. Then, we concatenate them with point positions along the sequence dimension. Motivated by CogVideoX [88], we apply the adaptive layer norm to positional tokens and physical tokens separately to facilitate the alignment between the two spaces:

$$\hat{\mathbf{P}}^f = \text{SelfAttn}(\text{AdaLN}([\mathbf{P}^f; \mathbf{cond}])) , \quad \forall f \in [1, F] \quad (4)$$

Temporal attention mainly aggregates information of the same point across all timesteps for temporal consistency. We also apply attention to the input point cloud \mathbf{P}_0 for better trajectory learning.

$$\hat{\mathbf{T}}_p = \text{SelfAttn}(\text{AdaLN}([\mathbf{T}_p])) , \quad \forall p \in [1, N] \quad (5)$$

where $\mathbf{T}_p = [\mathbf{x}_p^0, \mathbf{x}_p^1, \mathbf{x}_p^2, \dots, \mathbf{x}_p^F] \in \mathbb{R}^{(F+1) \times d}$.

4.1.3 Training Losses

We train a standard diffusion model in which we add Gaussian noise ϵ of different levels t to the entire point cloud sequence: $\mathcal{P}_t = \alpha_t \mathcal{P} + \sigma_t \epsilon$ and then feed the noisy point cloud sequence into the denoising network \mathcal{D} . We use the signal-prediction formulation of diffusion models: $\hat{\mathcal{P}} = \mathcal{D}(\mathcal{P}_t, t, c)$.

Diffusion Loss We use MSE loss between the predicted and ground truth signal given noise samples:

$$\mathcal{L}_{\text{diff}} = \mathbb{E}_{\mathcal{P} \sim q(\mathcal{P}|c), t \sim [1, T]} \|\mathcal{D}(\mathcal{P}_t; t, c) - \mathcal{P}\|_2^2 \quad (6)$$

Velocity Loss We regulate the velocity across two frames, similar to that used in MDM [74]:

$$\mathcal{L}_{\text{vel}} = \frac{1}{F-1} \sum_{f=1}^{F-1} \|(\mathcal{P}^{f+1} - \mathcal{P}^f) - (\hat{\mathcal{P}}^{f+1} - \hat{\mathcal{P}}^f)\|_2^2 \quad (7)$$

Physics Loss To enable the model to learn physics-plausible motion trajectories, we introduce a physics-based supervision to enforce physical plausibility for the elastic, plasticine and sand material from MPM. Specifically, we constrain the position and velocity of the predicted points to adhere to the deformation gradient update (Equation (3)) across frames:

$$\mathcal{L}_{\text{phys}} = \frac{1}{N(F-2)} \sum_{f=1}^{F-2} \sum_{p=1}^N \|\mathbf{F}_p^{f+1} - g(\hat{\mathbf{x}}_p^f) \mathbf{F}_p^f\|_2 \quad g(\hat{\mathbf{x}}_p^f) = \mathbf{I} + \Delta T \sum_i \hat{\mathbf{v}}_i^{f+1} \nabla N(\mathbf{x}_i - \hat{\mathbf{x}}_p^f)^\top \quad (8)$$

where \mathbf{F}_p^{f+1} and \mathbf{F}_p^f are the ground-truth deformation gradient between adjacent frames and $\hat{\mathbf{x}}_p^f \in \hat{\mathcal{P}}^f$ is the predicted position. To obtain an approximation of grid velocity $\hat{\mathbf{v}}_i^{f+1}$ in Equation (8), we perform one P2G and G2P step (Equation (2)) at each frame in training. This can be formulated as

$$\hat{\mathbf{v}}_i^{f+1} = \frac{\sum_p N_i(\hat{\mathbf{x}}_p^f) m_p (\hat{\mathbf{v}}_p^{f+1} + \mathbf{C}_p^f (\mathbf{x}_i - \hat{\mathbf{x}}_p^f))}{\sum_p N_i(\hat{\mathbf{x}}_p^f) m_p} \quad (9)$$

where \mathbf{C}_p^f is also from ground-truth and $\hat{\mathbf{v}}_p^{f+1} = (\hat{\mathbf{x}}_p^{f+2} - \hat{\mathbf{x}}_p^f) / (2\Delta T)$. Note that we ignore the stress term and use next-frame point velocity $\hat{\mathbf{v}}_p^{f+1}$ because it yields a more accurate approximation when the frame interval ΔT is much larger than the substep interval Δt for MPM simulation.

Boundary Loss To enforce the boundary condition of the ground, we add a penetration loss, preventing the points from passing through the surface:

$$\mathcal{L}_{\text{floor}} = \frac{1}{N} \sum_{f=1}^F \sum_{p=1}^N (\max(h - \hat{\mathbf{x}}_p^f, 0))^2 \quad (10)$$

Overall, our training loss is: $\mathcal{L} = \mathcal{L}_{\text{diff}} + \lambda_{\text{vel}} \mathcal{L}_{\text{vel}} + \lambda_{\text{phys}} \mathcal{L}_{\text{phys}} + \lambda_{\text{floor}} \mathcal{L}_{\text{floor}}$.



Figure 4: Qualitative comparison between our method and existing video generation methods.

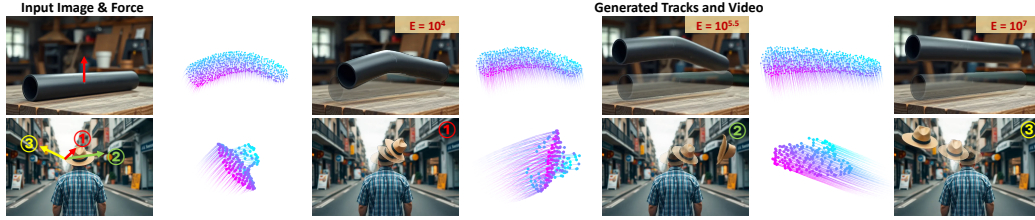


Figure 5: PhysCtrl generates videos of the same object under different physics parameters and forces.

4.2 Physics-grounded Image-to-Video Generation

Starting with a single image of 3D scene with objects, we first segment out [37] the objects and generate novel view images for each object. We then feed both the novel views and the segmented image into a multiview Gaussian reconstruction model [73] and extract a point cloud for the input objects. We then use our trajectory generative model to generate the dynamics of that point cloud. The generated 3D point trajectories are then projected to image space of the input camera viewpoint to obtain the motion trajectories of each pixel. The projected pixel trajectories can be directly used as conditioning signals for a pre-trained video generative model [24] to produce the final video.

5 Experiments

5.1 Evaluation on Image-to-Video Generation

Baselines We compare PhysCtrl with state-of-the-art controllable video generative models, including Wan2.1-I2V-14B [77], CogVideoX [88], DragAnything [83], ObjCtrl-2.5D [81]. The first two methods support image-to-video generation with text prompts. We use ChatGPT-4o to generate text prompts based on the direction of the object movement. The last two achieve controllable video generation with user-specified single-point trajectories. We use the trajectories of the drag point generated by our model to prompt the model.

Quantitative Evaluation Since we are the first method to inject physics prior into a video model, we utilize GPT-4o to evaluate three aspects of 12 generated videos in a 5-Likert score inspired by VideoPhy [2]: (1) Semantic Adherence (SA): how well the content and motion in the video match the description in the text prompt, especially the alignment with the force direction and position; (2) Physical commonsense (PC): whether the object’s motion follows intuitive, physically plausible dynamics given the applied force direction and position; (3) Video Quality (VQ): overall visual and temporal quality of the video. Results in Table 1 show that our method achieve the best results across all baselines. Results of user study can be found in the supplemental.

Qualitative Evaluation The qualitative results between our method and baselines can be found in Figure 4. CogVideoX-5B [88] and Wan2.1 [77] have strong generation ability and partly follow the

text prompts. However, they only use text prompts as conditions and lack precise control, thus, they cannot produce motions that fully reflect physics laws. For example, the *chair* in Figure 4 doesn’t move according to the force direction. DragAnything [83] uses purely 2D trajectories and cannot distinguish between camera motion and object motion, thus sometimes generating camera motions while objects remain static. More importantly, both DragAnything [83] and ObjCtrl2.5D [81] only use coarse trajectory as a condition and struggle to generate more complex motions, *e.g.*, the UFO case in Figure 4 that contains both rotations and depth change. In comparison, PhysCtrl produces physics-plausible videos that follow the given forces by generating physics-grounded 3D trajectories as a strong conditional signal to guide the superior generation capability of pretrained video generative models for video synthesis.

Table 1: Results of video evaluation.

	SA \uparrow	PC \uparrow	VQ \uparrow
DragAnything [83]	2.9	2.8	2.8
ObjCtrl [81]	1.5	1.3	1.4
Wan2.1 [77]	3.8	3.7	3.6
CogVideoX [88]	3.2	3.2	3.1
Ours	4.5	4.5	4.3

Table 2: Quantitative comparison on trajectory generation.

Method	vIoU \uparrow	CD \downarrow	Corr \downarrow
M2V [6]	24.92%	0.2160	0.1064
MDM [74]	53.78%	0.0159	0.0240
Ours	77.03%	0.0030	0.0016

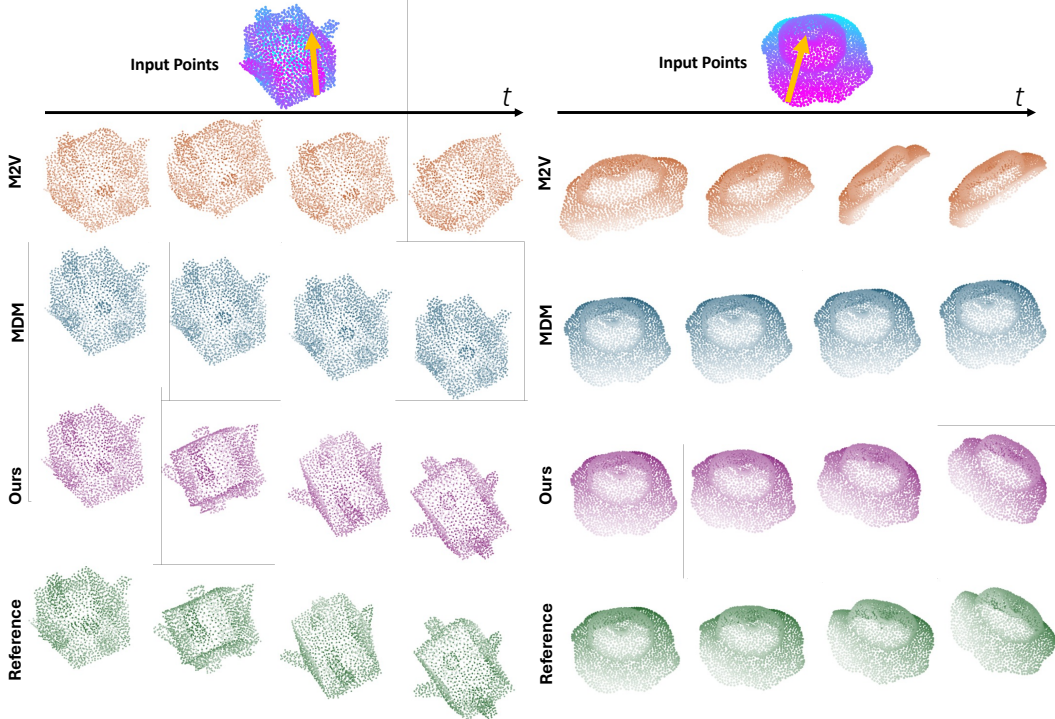


Figure 6: **Qualitative results:** Compared to baselines, our method enables high-quality and coherent generation of motion sequences from physics conditions and closely matches the reference.

Results on Varying Physical Conditions Since our trajectory generative model is conditioned on external forces and physics parameters, we can generate videos of the same object under varying conditions. As shown in Figure 5, we can change the Young’s modulus in elastic material to produce results with different deformations given the same force. The direction and amplitude of the force can also be adjusted to match the user’s desired motion. We found that Poisson’s ratio ν has negligible influence on the generated trajectories, similar to the findings in PhysDreamer [91].

5.2 Evaluation on Generative Dynamics

Baselines We compare our approach with existing methods that focus on generative dynamics, including Motion2VecSets [6] and MDM [74]. Motion2VecSets is a method for reconstructing sparse point cloud sequences; we eliminate the sparse point cloud condition and introduce physics conditions instead. MDM is primarily aimed at human motion generation, so we substitute human joints with point clouds and incorporate physics conditions as additional tokens. For computation efficiency,

we trained all baselines and ablations on our elastic subset of 160K objects that contains complex deformations for metrics comparison.

Evaluation Metrics Following [41, 6], we adopt volume Intersection over Union (vIoU), Chamfer Distance (CD) and L_2 -distance error for evaluation. vIoU measures the overlap between predicted and ground truth point clouds, CD measures the averaged per-point pairwise nearest neighbor distance between two point clouds, L_2 -distance is the Euclidean distance between two corresponding point clouds. Each metric is calculated at each timestep separately and averaged across all frames.

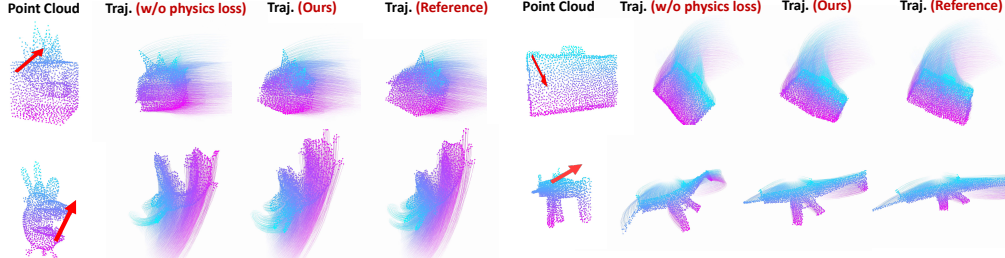


Figure 7: Comparison of using physics loss on trajectory generation. With physics loss, the results are more closely aligned with the reference.

Results Table 2 shows the quantitative comparison of our method with other baselines. Our method demonstrates the best performance over all metrics on the testing set. The qualitative comparison can be found in Figure 1. Our model achieves physics-grounded and consistent generation of motion trajectories. Motion2vecsets struggles to generate time-coherent motions because in our experiments, there is no sparse point cloud condition in their original setting. M2V struggles to generate coherent motions in our experiments. There are two potential reasons for this. Firstly, their model is originally designed for point cloud completion, but in our setting, there is no sparse point cloud condition. Prior work [92] also found that M2V does not work well in this situation. Secondly, their deformation latent is encoded frame-by-frame without temporal interaction. MDM can generate consistent motion sequences, but fails to capture detailed deformations because all points in a frame are projected into a single latent. The superiority of our method is based on our spatial-temporal attention block, which leverages explicit per-point correspondence.

Table 3: Ablation study on our trajectory generative model.

Method	vIoU \uparrow	CD \downarrow	Corr \downarrow
w/o spatial attention	28.88%	0.2700	0.1987
w/o temporal attention	54.92%	0.0664	0.0566
w/o physics loss	75.59%	0.0033	0.0018
Ours	77.03%	0.0030	0.0016

5.3 Ablation Study

The qualitative and quantitative results of the ablation study can be found in Figure 7 and Table 3. Our physics loss improved all the metrics and makes the results of our trajectory generation close to the ground truth. The physics loss aligns the updated deformation gradient with the ground truth and constrains the predicted positions.

6 Conclusion and Limitations

In this paper, we introduce PhysCtrl, a novel framework for physics-grounded video generation with physics parameters and force control. We design a diffusion model with spatial-temporal attention blocks and physics-based supervision to effectively and efficiently learn complex physical deformations directly on point cloud sequences. The generated motion trajectories can be used as a strong conditional signal for pre-trained video generative models. Our experiments demonstrate that PhysCtrl can generate physics-grounded dynamics and enable high-quality image-to-video generation results conditioned on external forces and physics parameters.

Our approach mostly focuses on single-object dynamics for four material types and does not cover all possible materials. We also does not model highly complex phenomena, such as multi-object coupling or intricate boundary conditions. Future work includes addressing these limitations and extending PhysCtrl to more diverse and complex physics phenomena in the real world.

References

- [1] Pymunk (2023), <https://pymunk.org>
- [2] Bansal, H., Lin, Z., Xie, T., Zong, Z., Yarom, M., Bitton, Y., Jiang, C., Sun, Y., Chang, K.W., Grover, A.: Videophy: Evaluating physical commonsense for video generation. arXiv preprint arXiv:2406.03520 (2024)
- [3] Bansal, H., Peng, C., Bitton, Y., Goldenberg, R., Grover, A., Chang, K.W.: Videophy-2: A challenging action-centric physical commonsense evaluation in video generation. arXiv preprint arXiv:2503.06800 (2025)
- [4] Blattmann, A., Dockhorn, T., Kulal, S., Mendelevitch, D., Kilian, M., Lorenz, D., Levi, Y., English, Z., Voleti, V., Letts, A., et al.: Stable video diffusion: Scaling latent video diffusion models to large datasets. arXiv preprint arXiv:2311.15127 (2023)
- [5] Burgert, R., Xu, Y., Xian, W., Pilarski, O., Clausen, P., He, M., Ma, L., Deng, Y., Li, L., Mousavi, M., 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. pp. 13–23 (2025)
- [6] Cao, W., Luo, C., Zhang, B., Nießner, M., Tang, J.: Motion2vecsets: 4d latent vector set diffusion for non-rigid shape reconstruction and tracking. In: CVPR. pp. 20496–20506 (2024)
- [7] Che, H., He, X., Liu, Q., Jin, C., Chen, H.: Gamegen-x: Interactive open-world game video generation. arXiv preprint arXiv:2411.00769 (2024)
- [8] Chen, B., Jiang, H., Liu, S., Gupta, S., Li, Y., Zhao, H., Wang, S.: Physgen3d: Crafting a miniature interactive world from a single image. arXiv preprint arXiv:2503.20746 (2025)
- [9] Chen, C., Dou, Z., Wang, C., Huang, Y., Chen, A., Feng, Q., Gu, J., Liu, L.: Vid2sim: Generalizable, video-based reconstruction of appearance, geometry and physics for mesh-free simulation. IEEE Conference on Computer Vision and Pattern Recognition (CVPR) (2025)
- [10] Chen, H., Xia, M., He, Y., Zhang, Y., Cun, X., Yang, S., Xing, J., Liu, Y., Chen, Q., Wang, X., et al.: Videocrafter1: Open diffusion models for high-quality video generation. arXiv preprint arXiv:2310.19512 (2023)
- [11] Chen, H., Zhang, Y., Cun, X., Xia, M., Wang, X., Weng, C., Shan, Y.: Videocrafter2: Overcoming data limitations for high-quality video diffusion models. In: CVPR. pp. 7310–7320 (2024)
- [12] Chen, R.T., Rubanova, Y., Bettencourt, J., Duvenaud, D.K.: Neural ordinary differential equations. NeurIPS **31** (2018)
- [13] Chu, M., Liu, L., Zheng, Q., Franz, E., Seidel, H.P., Theobalt, C., Zayer, R.: Physics informed neural fields for smoke reconstruction with sparse data. ACM TOG **41**(4), 1–14 (2022)
- [14] Decart, E., Campbell, S., McIntyre, Q., Chen, X., Quevedo, J.: Oasis: A universe in a transformer (2024)
- [15] Deitke, M., Liu, R., Wallingford, M., Ngo, H., Michel, O., Kusupati, A., Fan, A., Laforte, C., Voleti, V., Gadre, S.Y., et al.: Objaverse-xl: A universe of 10m+ 3d objects. Advances in Neural Information Processing Systems **36**, 35799–35813 (2023)
- [16] Deitke, M., Schwenk, D., Salvador, J., Weihs, L., Michel, O., VanderBilt, E., Schmidt, L., Ehsani, K., Kembhavi, A., Farhadi, A.: Objaverse: A universe of annotated 3d objects. In: CVPR. pp. 13142–13153 (2023)
- [17] Desbrun, M.: Smoothed particles: A new paradigm for animating highly deformable bodies. Computer Animation and Simulation/Springer Vienna (1996)
- [18] Erkoç, Z., Ma, F., Shan, Q., Nießner, M., Dai, A.: Hyperdiffusion: Generating implicit neural fields with weight-space diffusion. In: ICCV. pp. 14300–14310 (2023)
- [19] Feng, Y., Shang, Y., Feng, X., Lan, L., Zhe, S., Shao, T., Wu, H., Zhou, K., Su, H., Jiang, C., et al.: Elastogen: 4d generative elastodynamics. arXiv preprint arXiv:2405.15056 (2024)
- [20] Fu, X., Liu, X., Wang, X., Peng, S., Xia, M., Shi, X., Yuan, Z., Wan, P., Zhang, D., Lin, D.: 3dtrajmaster: Mastering 3d trajectory for multi-entity motion in video generation. arXiv preprint arXiv:2412.07759 (2024)

- [21] Geng, D., Herrmann, C., Hur, J., Cole, F., Zhang, S., Pfaff, T., Lopez-Guevara, T., Doersch, C., Aytar, Y., Rubinstein, M., et al.: Motion prompting: Controlling video generation with motion trajectories. arXiv preprint arXiv:2412.02700 (2024)
- [22] Gillman, N., Herrmann, C., Freeman, M., Aggarwal, D., Luo, E., Sun, D., Sun, C.: Force prompting: Video generation models can learn and generalize physics-based control signals. arXiv preprint arXiv:2505.19386 (2025)
- [23] Gong, S., Li, M., Feng, J., Wu, Z., Kong, L.: Diffuseq: Sequence to sequence text generation with diffusion models. ICLR (2023)
- [24] Gu, Z., Yan, R., Lu, J., Li, P., Dou, Z., Si, C., Dong, Z., Liu, Q., Lin, C., Liu, Z., et al.: Diffusion as shader: 3d-aware video diffusion for versatile video generation control. arXiv preprint arXiv:2501.03847 (2025)
- [25] He, H., Xu, Y., Guo, Y., Wetzstein, G., Dai, B., Li, H., Yang, C.: Cameractrl: Enabling camera control for text-to-video generation. arXiv preprint arXiv:2404.02101 (2024)
- [26] He, Y., Yang, T., Zhang, Y., Shan, Y., Chen, Q.: Latent video diffusion models for high-fidelity long video generation. arXiv preprint arXiv:2211.13221 (2022)
- [27] Ho, J., Chan, W., Saharia, C., Whang, J., Gao, R., Gritsenko, A., Kingma, D.P., Poole, B., Norouzi, M., Fleet, D.J., et al.: Imagen video: High definition video generation with diffusion models. arXiv preprint arXiv:2210.02303 (2022)
- [28] Ho, J., Jain, A., Abbeel, P.: Denoising diffusion probabilistic models. NeurIPS **33**, 6840–6851 (2020)
- [29] Ho, J., Salimans, T., Gritsenko, A.A., Chan, W., Norouzi, M., Fleet, D.J.: Video diffusion models. In: ICLR Workshop on Deep Generative Models for Highly Structured Data (2022), <https://openreview.net/forum?id=BBelR2NdZ5>
- [30] Hu, L.: Animate anyone: Consistent and controllable image-to-video synthesis for character animation. In: CVPR. pp. 8153–8163 (2024)
- [31] Hu, Y., Fang, Y., Ge, Z., Qu, Z., Zhu, Y., Pradhana, A., 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)
- [32] Hu, Y., Li, T.M., Anderson, L., Ragan-Kelley, J., Durand, F.: Taichi: a language for high-performance computation on spatially sparse data structures. ACM Transactions on Graphics (TOG) **38**(6), 1–16 (2019)
- [33] Jiang, C., Gast, T., Teran, J.: Anisotropic elastoplasticity for cloth, knit and hair frictional contact. ACM Transactions on Graphics (TOG) **36**(4), 1–14 (2017)
- [34] Jiang, C., Schroeder, C., Selle, A., Teran, J., Stomakhin, A.: The affine particle-in-cell method. ACM Transactions on Graphics (TOG) **34**(4), 1–10 (2015)
- [35] Jiang, C., Schroeder, C., Teran, J., Stomakhin, A., Selle, A.: The material point method for simulating continuum materials. In: Acm siggraph 2016 courses, pp. 1–52 (2016)
- [36] Jiang, Y., Yu, C., Xie, T., Li, X., Feng, Y., Wang, H., Li, M., Lau, H., Gao, F., Yang, Y., et al.: Vr-gs: A physical dynamics-aware interactive gaussian splatting system in virtual reality. In: ACM SIGGRAPH 2024 Conference Papers. pp. 1–1 (2024)
- [37] Kirillov, A., Mintun, E., Ravi, N., Mao, H., Rolland, C., Gustafson, L., Xiao, T., Whitehead, S., Berg, A.C., Lo, W.Y., et al.: Segment anything. In: ICCV. pp. 4015–4026 (2023)
- [38] Klár, G., Gast, T., Pradhana, A., Fu, C., Schroeder, C., Jiang, C., Teran, J.: Drucker-prager elastoplasticity for sand animation. ACM Transactions on Graphics (TOG) **35**(4), 1–12 (2016)
- [39] Kong, W., Tian, Q., Zhang, Z., Min, R., Dai, Z., Zhou, J., Xiong, J., Li, X., Wu, B., Zhang, J., et al.: Hunyuanvideo: A systematic framework for large video generative models. arXiv preprint arXiv:2412.03603 (2024)
- [40] Kugelstadt, T., Bender, J., Fernández-Fernández, J.A., Jeske, S.R., Löschner, F., Longva, A.: Fast corotated elastic sph solids with implicit zero-energy mode control. Proceedings of the ACM on Computer Graphics and Interactive Techniques **4**(3), 1–21 (2021)
- [41] Lei, J., Daniilidis, K.: Cadex: Learning canonical deformation coordinate space for dynamic surface representation via neural homeomorphism. In: CVPR. pp. 6624–6634 (2022)

- [42] Li, T., Bolkart, T., Black, M.J., Li, H., Romero, J.: Learning a model of facial shape and expression from 4d scans. *ACM TOG* **36**(6), 194–1 (2017)
- [43] Li, Z., Yu, H.X., Liu, W., Yang, Y., Herrmann, C., Wetzstein, G., Wu, J.: Wonderplay: Dynamic 3d scene generation from a single image and actions. *arXiv preprint arXiv:2505.18151* (2025)
- [44] Liang, J., Liu, R., Ozguroglu, E., Sudhakar, S., Dave, A., Tokmakov, P., Song, S., Vondrick, C.: Dreamitate: Real-world visuomotor policy learning via video generation. *arXiv preprint arXiv:2406.16862* (2024)
- [45] Liu, R., Wu, R., Van Hoorick, B., Tokmakov, P., Zakharov, S., Vondrick, C.: Zero-1-to-3: Zero-shot one image to 3d object. In: *ICCV*. pp. 9298–9309 (2023)
- [46] Liu, S., Ren, Z., Gupta, S., Wang, S.: Physgen: Rigid-body physics-grounded image-to-video generation. In: *ECCV*. pp. 360–378. Springer (2024)
- [47] Liu, T., Bargteil, A.W., O’Brien, J.F., Kavan, L.: Fast simulation of mass-spring systems. *ACM TOG* **32**(6), 1–7 (2013)
- [48] Long, X., Guo, Y.C., Lin, C., Liu, Y., Dou, Z., Liu, L., Ma, Y., Zhang, S.H., Habermann, M., Theobalt, C., et al.: Wonder3d: Single image to 3d using cross-domain diffusion. In: *CVPR*. pp. 9970–9980 (2024)
- [49] Loper, M., Mahmood, N., Romero, J., Pons-Moll, G., Black, M.J.: Smpl: A skinned multi-person linear model. In: *Seminal Graphics Papers: Pushing the Boundaries, Volume 2*, pp. 851–866 (2023)
- [50] Macklin, M., Müller, M.: Position based fluids. *ACM Transactions on Graphics (TOG)* **32**(4), 1–12 (2013)
- [51] Macklin, M., Müller, M., Chentanez, N.: Xpbd: position-based simulation of compliant constrained dynamics. In: *Proceedings of the 9th International Conference on Motion in Games*. pp. 49–54 (2016)
- [52] Mescheder, L., Oechsle, M., Niemeyer, M., Nowozin, S., Geiger, A.: Occupancy networks: Learning 3d reconstruction in function space. In: *CVPR*. pp. 4460–4470 (2019)
- [53] Mildenhall, B., Srinivasan, P., Tancik, M., Barron, J., Ramamoorthi, R., Ng, R.: Nerf: Representing scenes as neural radiance fields for view synthesis. In: *ECCV* (2020)
- [54] Modi, V., Sharp, N., Perel, O., Sueda, S., Levin, D.I.: Simplicits: Mesh-free, geometry-agnostic elastic simulation. *ACM TOG* **43**(4), 1–11 (2024)
- [55] Müller, M., Heidelberger, B., Hennix, M., Ratcliff, J.: Position based dynamics. *Journal of Visual Communication and Image Representation* **18**(2), 109–118 (2007)
- [56] Niemeyer, M., Mescheder, L., Oechsle, M., Geiger, A.: Occupancy flow: 4d reconstruction by learning particle dynamics. In: *ICCV*. pp. 5379–5389 (2019)
- [57] OpenAI: (2024), <https://openai.com/index/sora>
- [58] Peer, A., Gissler, C., Band, S., Teschner, M.: An implicit sph formulation for incompressible linearly elastic solids. In: *Computer Graphics Forum*. vol. 37, pp. 135–148. Wiley Online Library (2018)
- [59] Raissi, M., Perdikaris, P., Karniadakis, G.E.: Physics-informed neural networks: A deep learning framework for solving forward and inverse problems involving nonlinear partial differential equations. *Journal of Computational physics* **378**, 686–707 (2019)
- [60] Ram, D., Gast, T., Jiang, C., Schroeder, C., Stomakhin, A., Teran, J., Kavehpour, P.: A material point method for viscoelastic fluids, foams and sponges. In: *Proceedings of the 14th ACM SIGGRAPH/Eurographics Symposium on Computer Animation*. pp. 157–163 (2015)
- [61] Rombach, R., Blattmann, A., Lorenz, D., Esser, P., Ommer, B.: High-resolution image synthesis with latent diffusion models. In: *CVPR*. pp. 10684–10695 (2022)
- [62] Romero, J., Tzionas, D., Black, M.J.: Embodied hands: Modeling and capturing hands and bodies together. *ACM TOG* **36**(6) (2017)
- [63] Saharia, C., Chan, W., Saxena, S., Li, L., Whang, J., Denton, E.L., Ghasemipour, K., Gontijo Lopes, R., Karagol Ayan, B., Salimans, T., et al.: Photorealistic text-to-image diffusion models with deep language understanding. *NeurIPS* **35**, 36479–36494 (2022)

- [64] Sanchez-Gonzalez, A., Godwin, J., Pfaff, T., Ying, R., Leskovec, J., Battaglia, P.: Learning to simulate complex physics with graph networks. In: International conference on machine learning. pp. 8459–8468. PMLR (2020)
- [65] Shi, H., Xu, H., Huang, Z., Li, Y., Wu, J.: Robocraft: Learning to see, simulate, and shape elasto-plastic objects in 3d with graph networks. *The International Journal of Robotics Research* **43**(4), 533–549 (2024)
- [66] Shue, J.R., Chan, E.R., Po, R., Ankner, Z., Wu, J., Wetzstein, G.: 3d neural field generation using triplane diffusion. In: CVPR. pp. 20875–20886 (2023)
- [67] Song, J., Meng, C., Ermon, S.: Denoising diffusion implicit models (2020)
- [68] Song, Y., Ermon, S.: Generative modeling by estimating gradients of the data distribution. *NeurIPS* **32** (2019)
- [69] Song, Y., Sohl-Dickstein, J., Kingma, D.P., Kumar, A., Ermon, S., Poole, B.: Score-based generative modeling through stochastic differential equations. In: ICLR (2020)
- [70] Stomakhin, A., Schroeder, C., Chai, L., Teran, J., Selle, A.: A material point method for snow simulation. *ACM Transactions on Graphics (TOG)* **32**(4), 1–10 (2013)
- [71] Tan, X., Jiang, Y., Li, X., Zong, Z., Xie, T., Yang, Y., Jiang, C.: Physmotion: Physics-grounded dynamics from a single image. *arXiv preprint arXiv:2411.17189* (2024)
- [72] Tang, J., Xu, D., Jia, K., Zhang, L.: Learning parallel dense correspondence from spatio-temporal descriptors for efficient and robust 4d reconstruction. In: CVPR. pp. 6022–6031 (2021)
- [73] Tang, J., Chen, Z., Chen, X., Wang, T., Zeng, G., Liu, Z.: Lgm: Large multi-view gaussian model for high-resolution 3d content creation. In: ECCV. pp. 1–18. Springer (2024)
- [74] Tevet, G., Raab, S., Gordon, B., Shafir, Y., Cohen-or, D., Bermano, A.H.: Human motion diffusion model. In: ICCV (2023)
- [75] Valevski, D., Leviathan, Y., Arar, M., Fruchter, S.: Diffusion models are real-time game engines. *arXiv preprint arXiv:2408.14837* (2024)
- [76] Voleti, V., Yao, C.H., Boss, M., Letts, A., Pankratz, D., Tochilkin, D., Laforte, C., Rombach, R., Jampani, V.: Sv3d: Novel multi-view synthesis and 3d generation from a single image using latent video diffusion. In: ECCV. pp. 439–457. Springer (2024)
- [77] Wang, A., Ai, B., Wen, B., Mao, C., Xie, C.W., Chen, D., Yu, F., Zhao, H., Yang, J., Zeng, J., Wang, J., Zhang, J., Zhou, J., Wang, J., Chen, J., Zhu, K., Zhao, K., Yan, K., Huang, L., Feng, M., Zhang, N., Li, P., Wu, P., Chu, R., Feng, R., Zhang, S., Sun, S., Fang, T., Wang, T., Gui, T., Weng, T., Shen, T., Lin, W., Wang, W., Wang, W., Zhou, W., Wang, W., Shen, W., Yu, W., Shi, X., Huang, X., Xu, X., Kou, Y., Lv, Y., Li, Y., Liu, Y., Wang, Y., Zhang, Y., Huang, Y., Li, Y., Wu, Y., Liu, Y., Pan, Y., Zheng, Y., Hong, Y., Shi, Y., Feng, Y., Jiang, Z., Han, Z., Wu, Z.F., Liu, Z.: Wan: Open and advanced large-scale video generative models. *arXiv preprint arXiv:2503.20314* (2025)
- [78] Wang, X., Zhu, Z., Huang, G., Chen, X., Zhu, J., Lu, J.: Drivedreamer: Towards real-world-drive world models for autonomous driving. In: ECCV. pp. 55–72. Springer (2024)
- [79] Wang, Y., Tang, S., Chu, M.: Physics-informed learning of characteristic trajectories for smoke reconstruction. In: ACM SIGGRAPH 2024 Conference Papers. pp. 1–11 (2024)
- [80] Wang, Y., He, J., Fan, L., Li, H., Chen, Y., Zhang, Z.: Driving into the future: Multiview visual forecasting and planning with world model for autonomous driving. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 14749–14759 (2024)
- [81] Wang, Z., Lan, Y., Zhou, S., Loy, C.C.: Objctrl-2.5 d: Training-free object control with camera poses. *arXiv preprint arXiv:2412.07721* (2024)
- [82] Wu, R., Gao, R., Poole, B., Trevithick, A., Zheng, C., Barron, J.T., Holynski, A.: Cat4d: Create anything in 4d with multi-view video diffusion models. *arXiv preprint arXiv:2411.18613* (2024)
- [83] Wu, W., Li, Z., Gu, Y., Zhao, R., He, Y., Zhang, D.J., Shou, M.Z., Li, Y., Gao, T., Zhang, D.: Draganything: Motion control for anything using entity representation. In: ECCV. pp. 331–348. Springer (2024)
- [84] Xie, T., Zhao, Y., Jiang, Y., Jiang, C.: Physanimator: Physics-guided generative cartoon animation. *arXiv preprint arXiv:2501.16550* (2025)

- [85] Xie, T., Zong, Z., Qiu, Y., Li, X., Feng, Y., Yang, Y., Jiang, C.: Physgaussian: Physics-integrated 3d gaussians for generative dynamics. In: CVPR. pp. 4389–4398 (2024)
- [86] Xing, J., Xia, M., Zhang, Y., Chen, H., Yu, W., Liu, H., Liu, G., Wang, X., Shan, Y., Wong, T.T.: Dynamicrafter: Animating open-domain images with video diffusion priors. In: ECCV. pp. 399–417. Springer (2024)
- [87] Yang, C., Gao, W., Wu, D., Wang, C.: Learning to simulate unseen physical systems with graph neural networks. arXiv preprint arXiv:2201.11976 (2022)
- [88] 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. arXiv preprint arXiv:2408.06072 (2024)
- [89] Zhang, K., Li, B., Hauser, K., Li, Y.: Adaptigraph: Material-adaptive graph-based neural dynamics for robotic manipulation. In: Proceedings of Robotics: Science and Systems (RSS) (2024)
- [90] Zhang, L., Wang, Z., Zhang, Q., Qiu, Q., Pang, A., Jiang, H., Yang, W., Xu, L., Yu, J.: Clay: A controllable large-scale generative model for creating high-quality 3d assets. ACM TOG **43**(4), 1–20 (2024)
- [91] Zhang, T., Yu, H.X., Wu, R., Feng, B.Y., Zheng, C., Snavely, N., Wu, J., Freeman, W.T.: Physdreamer: Physics-based interaction with 3d objects via video generation. In: ECCV. pp. 388–406. Springer (2024)
- [92] Zhang, X., Li, N., Dai, A.: Dnf: Unconditional 4d generation with dictionary-based neural fields. arXiv preprint arXiv:2412.05161 (2024)
- [93] Zhong, L., Yu, H.X., Wu, J., Li, Y.: Reconstruction and simulation of elastic objects with spring-mass 3d gaussians. In: ECCV. pp. 407–423. Springer (2024)
- [94] Zhou, W., Dou, Z., Cao, Z., Liao, Z., Wang, J., Wang, W., Liu, Y., Komura, T., Wang, W., Liu, L.: Emdm: Efficient motion diffusion model for fast and high-quality motion generation. In: ECCV. pp. 18–38. Springer (2024)
- [95] Zhu, S., Chen, J.L., Dai, Z., Dong, Z., Xu, Y., Cao, X., Yao, Y., Zhu, H., Zhu, S.: Champ: Controllable and consistent human image animation with 3d parametric guidance. In: ECCV. pp. 145–162. Springer (2024)
- [96] Zienkiewicz, O.C., Taylor, R.L., Nithiarasu, P., Zhu, J.: The finite element method, vol. 3. Elsevier (1977)
- [97] Zuffi, S., Kanazawa, A., Jacobs, D.W., Black, M.J.: 3d menagerie: Modeling the 3d shape and pose of animals. In: CVPR. pp. 6365–6373 (2017)

The supplementary material covers the following sections: Implementation Details Appendix **A**, User study Appendix **B**, Physics Parameter Estimation Appendix **C**, Results Appendix **D**, Societal Impacts Appendix **E**, Data and Model Safeguards Appendix **F**. **We also encourage readers to refer to our supplementary videos for demonstrations of animatable results..**

A Implementation Details

Dataset. To make our model handle diverse objects and motion trajectories, we generate data using physics simulation using high-quality 3D objects selected from ObjaverseXL [16, 15]. We simulate animations for each object with the MPM simulator [35] as the ground-truth. We use a fixed number of simulated points $N = 2048$ (uniformly sampled on the faces of the mesh) and frames $F = 24$ to align with our model’s input. For data augmentation, we randomly rotate the object around y -axis and add noise $\epsilon_p^{aug} \sim \mathcal{N}(0, 0.01^2)$ to each sampled initial point. Our whole dataset contains 550K objects, including 150K elastic objects of different drag force directions, 100K objects of gravity across elastic, sand, plasticine and rigid respectively. For the simulated animation of varying drag force, we randomly sample a constant force \mathbf{f} , a drag point $\mathbf{D} \in \mathbf{P}_0$ and physics parameters $E \in [10^4, 10^7]$, $\nu \in [0.05, 0.45]$. The force \mathbf{f} has an outward direction of the object surface and a magnitude between $0.02G$ and $0.3G$ in total (G is the gravity of the whole object) and is only applied to points close to the drag point \mathbf{D} .

Training For metric comparison and ablation, we train our base model on the 150K elastic subset that contains different force and physical parameters with 6 layers and 256 latent size on 8 NVIDIA L40 GPUs with 48GB GPU memory for 60K iterations with a total batch size of 32, which takes about 30 hours. We randomly leave out 100 animations from this dataset as the test set and keep the remaining ones for training. We train a large model of different materials with 12 layers and a 512 latent size on all 550K data with the same iterations and batch size, which takes about 80 hours. We use AdamW optimizer with betas (0.9, 0.999) and a learning rate of $1e-4$ with a cosine schedule and a warmup of 100 steps. We clip the gradient with the maximum norm of 1.0 and train with bfloat16 precision. We use a 25-step DDIM scheduler for sampling, which takes about 1 second for the base model and 3 seconds for the large model.

Image-to-3D Pipeline We use SAM [37] to segment the object in the input image and run SV3D [76] to generate 20 novel-view images of that object with orbit camera poses, from which we pick three images with azimuth (90° , 180° , 270°) relative to the input and send them together with the input into LGM [73] for 3D Gaussian reconstruction. We then convert the 3D Gaussians to a plain point cloud and sample N points using farthest point sampling (FPS) for trajectory generation.

GPT-4o Evaluation We prompt GPT-4o with the following prompt to use it for evaluation:

You are tasked with evaluating the quality of image-to-video generation produced by a model.
 For each test case, you will be given: 1. A text prompt describing a single object and a force applied to it. The force’s position and direction are visualized as a red arrow in the input image. 2. An input image of the object. 3. Five sets of 10 evenly spaced frames-each set corresponds to a video generated by a different model from the same input.
 Please evaluate this video based on the following three criteria using a 5-point Likert scale (1 = poor, 5 = excellent):
 - Semantic Adherence: How well the content and motion in the video match the description in the text prompt, especially the alignment with the force direction and position. Note that the video should starts with the input image.
 - Physical Commonsense: Whether the object’s motion follows intuitive, physically plausible dynamics given the applied force direction and position.
 - Video Quality: The overall visual and temporal quality of the video (note that static or nearly-static sequences are less preferred).
 Provide your evaluation for each video strictly in the following one-line format:
 Video i, Semantic Adherence score, Physical Commonsense score, Video Quality score

B User Study

We conducted a user study to evaluate the physics plausibility and overall quality of the videos generated by our model and other baselines. The study consisted of 12 questions, each including an input image with the force location and direction marked on the image, a text prompt describing the image and applied force, and generated video results produced by five different methods. The users are asked to carefully observe the videos and evaluate them from two aspects: (1) **Physics plausibility**: select the one that best matches the force direction (red arrow) and corresponding text prompt. The force and text prompt are assumed to match each other. (2) **Overall Video quality**: Select the one that has the best visual and temporal quality.

We received a total of 35 responses (35×12) and computed the percentage of times each method was selected as the best-performing video for each question. The results are summarized in Table 4, showing the preference rates for each method. The findings indicate that our model consistently outperforms baseline methods in terms of both physics plausibility and video quality. Although Wan

Table 4: Results of user study.

	Ours	CogVideoX	Wan	DragAnything	ObjCtrl2.5D
Physics Plausibility	81.0%	5.5%	10.2%	1.2%	2.1%
Video Quality	66.0%	6.2%	18.3%	4.5%	5.0%

Table 5: Mean Absolute Error (MAE) of Young’s Modulus on physics parameter estimation.

Method	Runtime (min.)	MAE of $\log_{10}(E)$
Ours	2	0.506
Diff. MPM (5 iters)	20	0.439
Diff. MPM (15 iters)	60	0.394

received the second-best video quality, some of these high-quality videos suffer from low physics plausibility.

C Physics Parameter Estimation

Our trained trajectory generation model learns the conditional distribution of physically plausible motion trajectories, so it can also be used for inverse problems, *i.e.*, to estimate the condition c given ground truth trajectories \mathcal{P} . The intuition is that **a c that is closer to the ground truth will introduce less discrepancy between the denoised trajectories and ground truth trajectories**. To this end, we define an energy function that measures how well the model can denoise a noisy version of \mathcal{P}_t under that condition:

$$\mathcal{E}(c) = \mathbb{E}_{t \sim [1, T]} \|\mathcal{P}_t - \mathcal{D}(\mathcal{P}_t; t, c)\|^2, \quad (11)$$

During optimization, the denoiser \mathcal{D} is frozen and only c is optimizable. We add random noise to the ground truth trajectory and feed it into the trained network to denoise. The gradient of the energy function will be backpropagated to optimize c .

We simulate 15 trajectories for elastic materials to test our physics parameter estimation pipeline. We compare our method with differentiable MPM [32], which needs to accumulate gradients over hundreds of substeps for one backward pass (costing more than 3min compared to 0.1s for ours). Table 5 shows that our method only takes about 2 minutes while achieving relatively good results, which also **demonstrates that our trained diffusion model captures physics-plausible motion trajectories**.

D More Results

More results of our method and baseline comparisons can be found in Figure 8. **We strongly encourage the readers to look at our video for better comparison, as isolated frames cannot fully represent the physical dynamics well.**

E Societal Impacts

Positive Impacts Our method integrates physically grounded simulation signals into video generative models, offering new avenues for controllable and physically plausible video synthesis. These can support people from amateurs to filmmakers and designers in rapidly prototyping ideas with accurate physical behavior, democratizing access to high-fidelity visual tools.

Negative Impacts High-fidelity generative models, especially when conditioned on physical signals, may be misused for creating deceptive content such as realistic yet fabricated disaster footage or physically plausible fake videos. This poses risks for misinformation and erosion of public trust. Although our approach enhances physical plausibility, it is important to note that the generated outputs are not real-world occurrences.

F Data and Model Safeguards

Given the dual-use nature of video generation models, we recognize that our pretrained model could be misused to generate deceptive, physically plausible videos for misinformation. As such, we will implement appropriate safeguards to support controlled access when we release our model, including: (1) requiring users to agree to usage guidelines and restrictions, (2) distributing the model under a research-only license, (3) investigating automatic safety filters that can flag potentially harmful uses. These steps aim to reduce the risk of malicious or unintended applications while still supporting reproducible research.

Our training data consists exclusively of synthetic point cloud trajectories representing object motion under simulated physics. These datasets contain no images, videos, or human-related content, and thus should pose no risk of visual misinformation, privacy violations, or unsafe content. All point clouds are generated in simulation environments and contain only geometric and physical information about object movement.

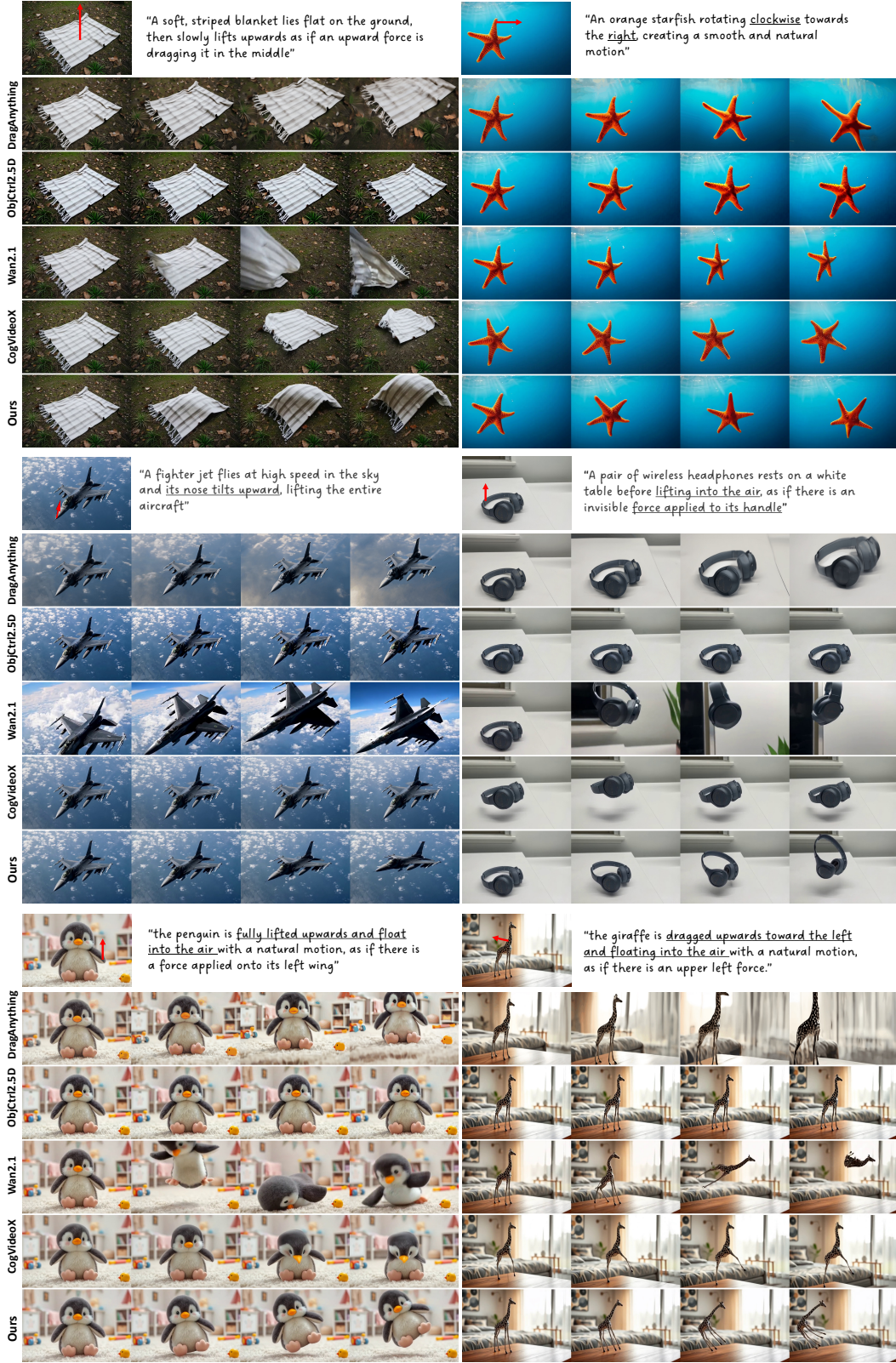


Figure 8: More qualitative comparison between our method and baselines.