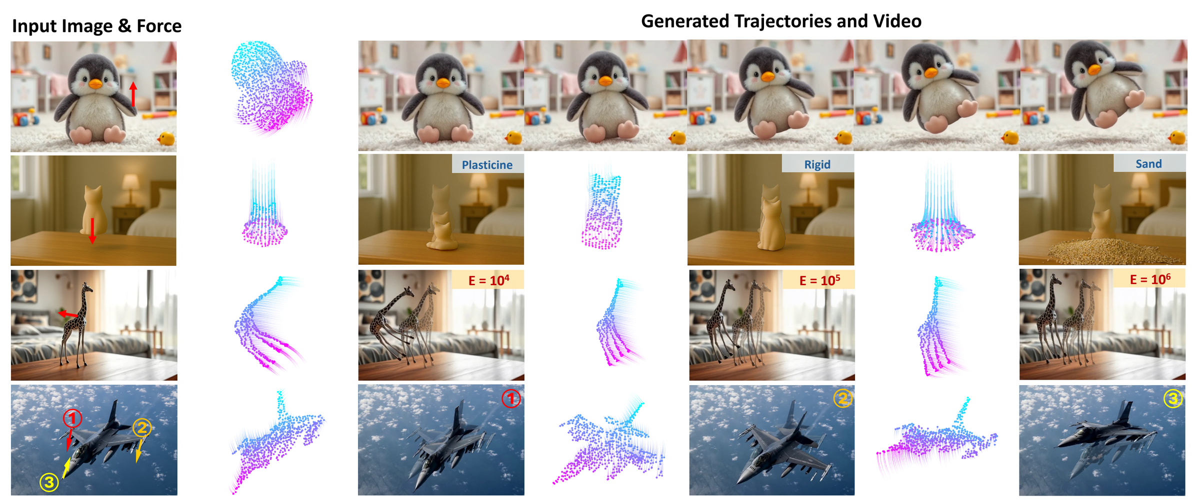
PhysCtrl: Generative Physics for Controllable and Physics-Grounded Video Generation

Chen Wang*, Chuhao Chen*, Yiming Huang, Zhiyang Dou, Yuan Liu, Jiatao Gu, Lingjie Liu

NeurIPS 2025 · Univ. of Penn, HKU, HKUST

TL,DR: We train a trajectory generative model conditioned on force and object material to achieve physics-grounded and controllable video generation.



Introduction and Motivation

Problems with Existing Video Generative Models:

1). **Controllability**: Lack control of what direction the object should go and the magnitude of movement

2). **Physics plausibility**: Object motion doesn't always follow physical laws, e.g. fall with gravity

Motivation: Leverage both motion priors from physical simulators and the generative ability of video models.

- Prior works (Motion Prompting / DAS) have shown that point trajectories can be used as a condition signal to drive pretrained video generative models
- Point tracks can generalize to different objects, various materials and dynamics

Problems of using traditional simulators:

1). Many "irrelevant" hyper-parameters like grid size, frame dt etc., unfriendly to amateurs

2). Tradeoff between robustness and speed because of numerical solvers

3). Different materials have to switch to different simulators

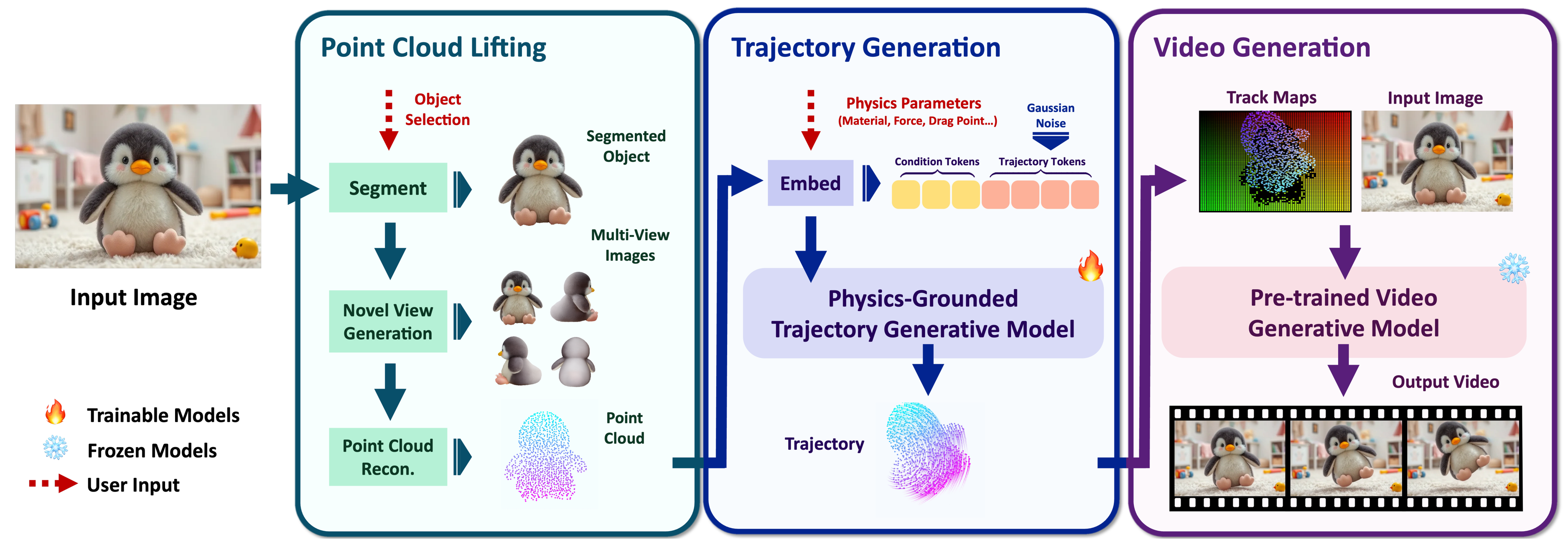
4). Speed is extremely slow for inverse problems due to the gradient propagation of many substeps

Our Solution: *Use a neural network as simulator!*

1). Collect a large-scale synthetic dataset of 550K object animations, spanning **elastic, sand, plasticine, and rigid** materials, using physics simulators.

2). Develop a diffusion-based point trajectory generative model equipped with a **spatiotemporal attention** mechanism and **physics-based constraint**.

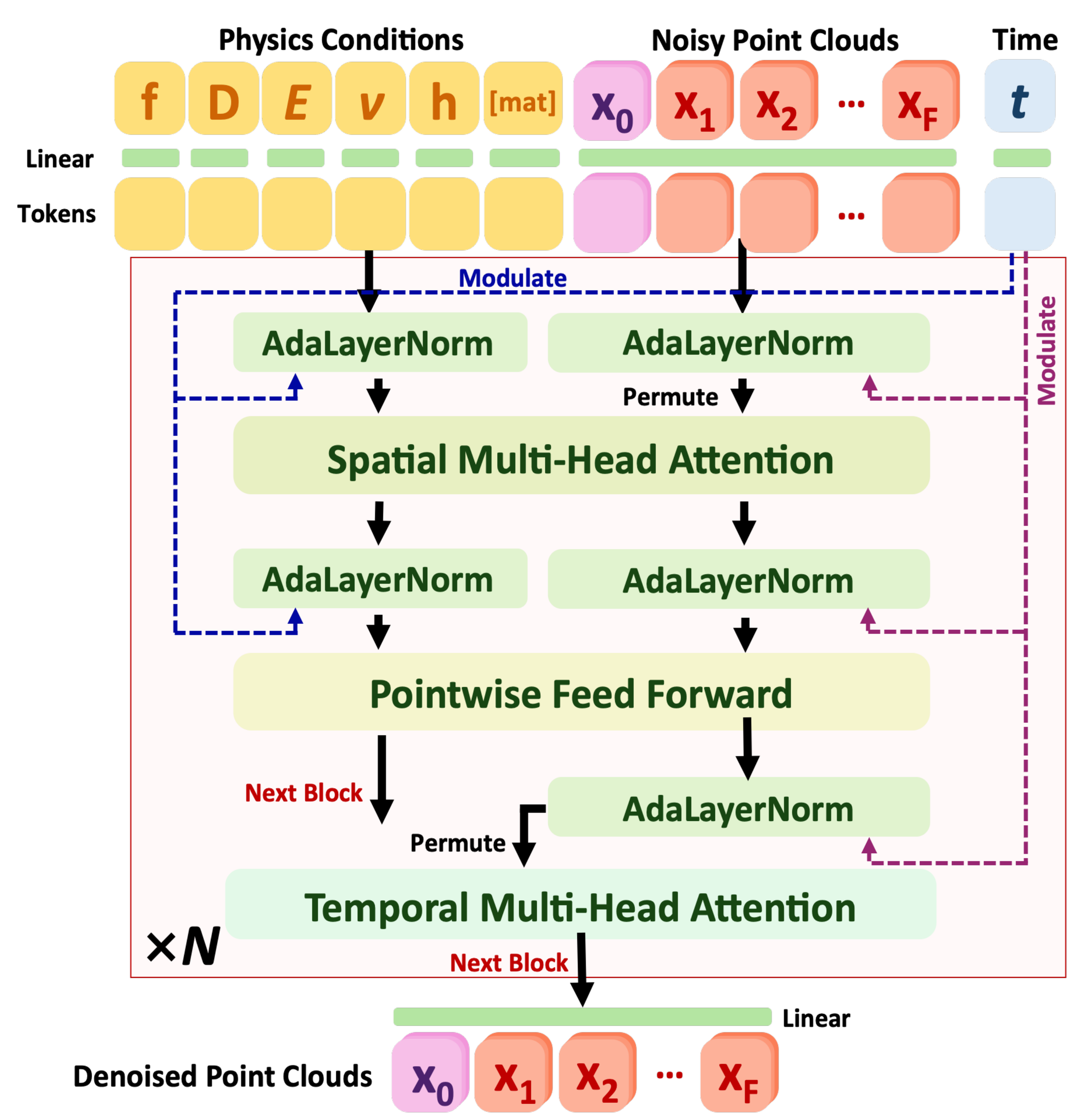
Method



Overview

From a single image, we first lift the object into 3D points and then generate motion tracks conditioned on physical parameters and external forces for video generation.

Trajectory Generation



Input: initial 3D point cloud, physical conditions (force location and direction, materials)
Output: Point cloud sequences at future frames

Tokenization: project points and physical conditions to two sets of tokens:

$$\mathbf{cond} = \mathrm{MLP}_{\mathrm{phys}}([\mathbf{f}; \mathbf{D}; \{E, \nu\}, h, [\mathrm{mat}]])$$

Spatial-Temporal Attention

Spatial: each point attends to points at the same frame

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

Temporal: each point attends its counterpart across frames

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

*Note: Spatial-temporal attention is more effective and efficient than full attention, as it **leverages pointwise correspondence** and **mimics the simulation process** (integrating information from neighboring points across space, then propagating forward across time)*

Train Losses

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

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

Boundary: $\mathcal{L}_{\mathrm{floor}} = \frac{1}{N}\sum_{f=1}^{F}\sum_{p=1}^{N} \left(\max(h - \hat{\mathbf{x}}_p^f, 0)\right)^2$

Physics loss based on Deformation Gradient:

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

Image-to-Video Generation

We use **multi-view generation** to reconstruct the input object into 3D points. Our trajectory generation model will generate future frames given force and materials. The generated 3D point trajectories are then **projected to the image space** of the viewpoint of input image to obtain the 2D motion trajectories.

Physics Parameter Estimation (Inversion problem)

Given known trajectory, estimate the input parameters.

Key idea: Parameters close to the ground truth will make the model generate trajectory closer to the GT trajectory

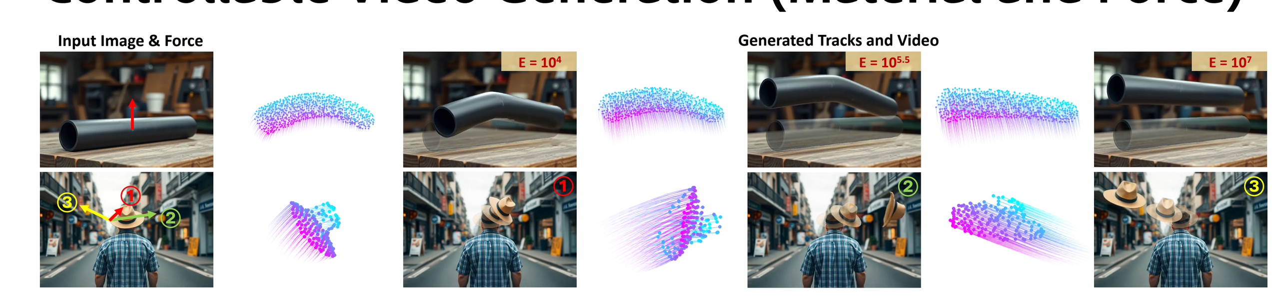
Use the following energy function to infer our trained model and optimize:

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

Results

Comparison with Existing Video Models

Controllable Video Generation (Material and Force)



Evaluation of Video Generation

We prompt GPT-4o to give 5-Likert Score for each model on Semantic Adherence (SA), Physical Commonsense (PC) and Visual Quality (VQ)

	SA↑	PC↑	VQ↑
DragAnything [89]	2.9	2.8	2.8
ObjCtrl [87]	1.5	1.3	1.4
Wan2.1 [82]	3.8	3.7	3.6
CogVideoX [94]	3.2	3.2	3.1
Ours	**4.5**	**4.5**	**4.3**

Evaluation of Trajectory Generation

We evaluate on geometry-based metrics.

Method	vIoU↑	CD↓	Corr↓
M2V [6]	24.92%	0.2160	0.1064
MDM [79]	53.78%	0.0159	0.0240
Ours	**77.59%**	**0.0028**	**0.0015**

Physical 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

Project Page, Video and Code