FORCE-CONDITIONED VIDEO GENERATION BY DECOUPLING MOTION-STATE PREDICTION AND VIDEO SYNTHESIS

Anonymous authors
Paper under double-blind review

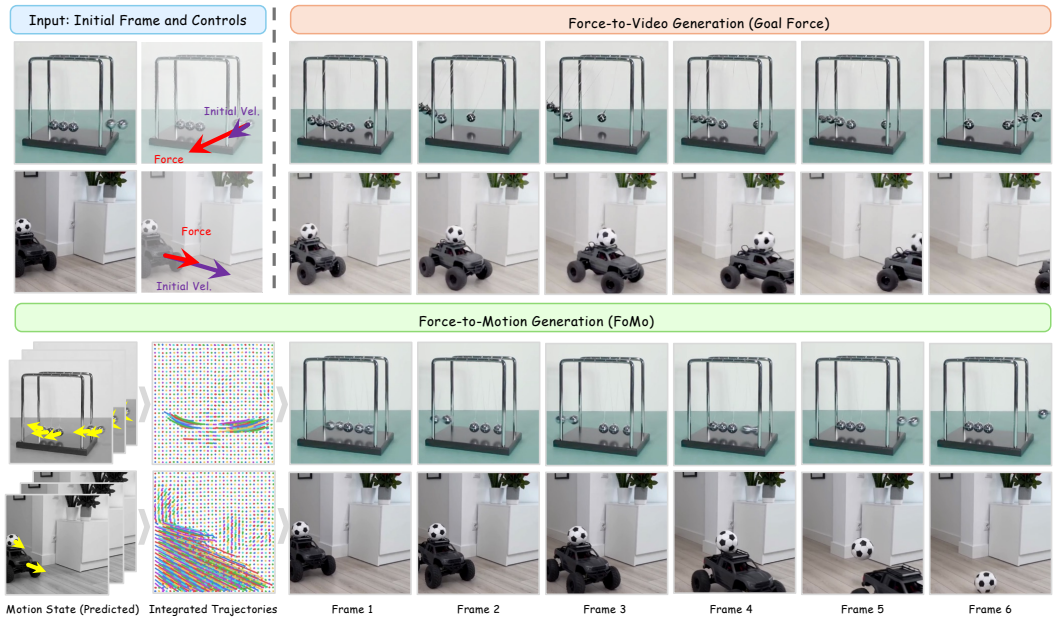

Figure 1: Direct force-to-video methods learn the force response through a video-generation objective without an explicit motion target. Different motions can produce similar frame appearances, but varying directions and speeds remain distinguishable in velocity space. FoMo therefore predicts initial-frame-indexed velocity and visibility maps, then integrates the velocities into object trajectories before video synthesis.

ABSTRACT

Interacting with the physical world using video diffusion models requires modeling how objects move under force control. Existing methods often map force controls directly to video. Without an explicit motion target, they learn force responses only through per-frame visual appearance. Appearance-based prediction may fail to capture the underlying physics, as different motion directions or strengths can appear similar at small motion amplitudes. Consequently, a generated video may look realistic without following the intended control. To improve physical plausibility and controllability, we propose FoMo, a generation scheme that separates motion-state prediction from video synthesis. Specifically, given an initial frame, relative force control, and initial velocity, instead of directly synthesizing the output video, FoMo first predicts initial-frame-indexed velocity and visibility maps. Velocity explicitly represents direction and magnitude, making motion changes direct training targets. In the next step, FoMo integrates the velocities into trajectories and passes them with the visibility maps to a pretrained trajectory-conditioned video generator. This design has two advantages: (1) direct motion supervision and (2) a generator focused only on visual synthesis. We additionally derive relative 2D controls from motion changes in real videos and use semantic physical priors for context. These controls do not represent exact 3D forces. Controlled ablations support explicit velocity prediction and real-video training. In a pairwise study with 100 participants, FoMo is preferred over Goal Force fine-tuned on the same training dataset.

Reviewers: please read the Reviewer Guidelines (iclr.cc/Conferences/2027/ReviewerGuidelines) and the AI Policy for Reviewers (iclr.cc/Conferences/2027/AIPolicyForReviewers). If you used AI to expand, edit, or polish your review, please provide the input text to the LLM. Better still, consider skipping the LLM and submitting your original text: we, and the authors, are much more interested in your unedited thoughts than in what an LLM has to say. AI-assisted or not, you are putting your name and reputation behind your review: LLM-generated falsehoods, hallucinations or misrepresentations are subject to disciplinary action, which may include desk-rejecting all papers you have authored.

1 INTRODUCTION

Video generation has achieved remarkable progress in synthesizing videos from text prompts, initial frames, and other visual conditions (Blattmann et al., 2023; Ho et al., 2022; Guo et al., 2023; Shi et al., 2024; Wan et al., 2025; Ma et al., 2026). The strong visual priors of these models make them suitable for rendering future frames, but interactive generation also requires a model of how the scene should move under control. Beyond visual synthesis, these advances have therefore raised increasing interest in using video generative models as world models that predict how a scene evolves under external interaction (Xie et al., 2024) and control (Agarwal et al., 2025; Wang et al., 2025b; Team et al., 2026; Wang et al., 2025a). Among these control mechanisms, force-based control is important because it specifies an interaction that should induce motion, rather than prescribing the desired visual trajectory (Burgert et al., 2025; Chu et al., 2025). Therefore, in this work, we study **force-conditioned video generation**. Given an initial frame and relative 2D controls for force and initial velocity, the model must infer the resulting object motion and synthesize high-fidelity future frames.

Recent methods such as Force Prompting and Goal Force extend video generation models to force-conditioned settings (Gillman et al., 2025; 2026). They construct paired force-video data, encode force signals as moving Gaussian blobs, and inject the resulting control into pretrained video diffusion models through ControlNet (Zhang et al., 2023; Chu et al., 2023). These methods predict RGB video directly and do not expose motion as an intermediate training target. Therefore, their force response is learned only through visual changes across frames. However, because force conditions with different directions or strengths may produce similar visual appearance, a model trained directly on visual appearance may fail to follow the intended force control. Their simulated training data introduce a separate limitation: coverage is set by manually configured objects, scenes, and parameter ranges (Gillman et al., 2025; 2026).

To make motion supervision explicit, we propose FoMo, a two-stage force-conditioned video generation model. It first predicts initial-frame-indexed velocity and visibility maps and integrates the velocities into trajectories. A pretrained trajectory-conditioned video generator then takes the initial frame, semantic prompt, trajectories, and visibility maps as conditions and synthesizes the final RGB video. This separation makes motion a direct training target while retaining the generator's visual prior. In velocity space, the direction and magnitude of each point are explicit, so changes that are subtle in RGB frames become clear learning targets. We choose velocity rather than acceleration as the prediction target because temporal differencing makes acceleration more sensitive to tracking noise near contact and collision. Also, for real-video training, the observed initial velocity change serves only as the relative control, while the full velocity sequence remains the prediction target. This construction allows FoMo to learn from both real and synthetic data and cover more objects and motion types (Appendix Fig. 5). Like Force Prompting and Goal Force, FoMo uses relative image-plane controls rather than calibrated world-space forces. Accordingly, we evaluate control consistency and visible physical plausibility rather than metric force recovery.

Additionally, to better model object motion, we design a motion state consisting of dense velocity and visibility maps over time, indexed by initial-frame points. The velocity maps track each point's motion, while the visibility maps indicate whether it remains visible. Unlike optical flow (Teed & Deng, 2020; Morimitsu et al., 2025), which describes adjacent-frame motion, this state maintains a common index throughout the video. We integrate the velocities into trajectories and pass them with the visibility maps to a pretrained trajectory-conditioned video generator.

Extensive evaluations support the effectiveness of our design. Our main contributions are:

- We formulate force-conditioned generation with initial-frame-indexed 2D velocity and visibility targets, enabling real-video supervision without 3D reconstruction or metric material parameters.
- We introduce FoMo, a diffusion network that predicts initial-frame-indexed velocity and visibility maps under sparse force control and integrates the velocities into trajectories for video synthesis.
- We broaden training with about 200K real-video clips and semantic physical priors, and evaluate control consistency and visual quality in complex dynamic scenes.

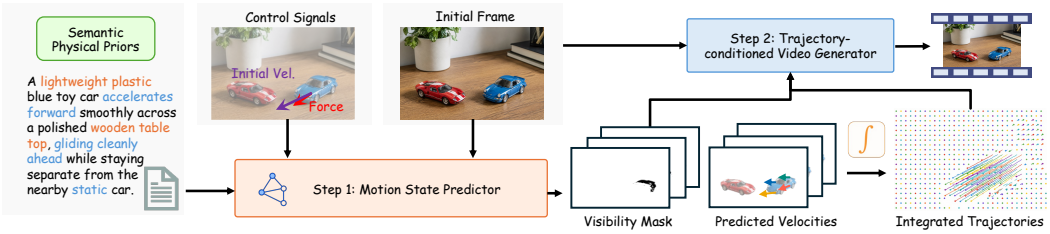

Figure 2: Framework of FoMo. Given an initial frame, relative force control, and initial velocity, the motion predictor estimates velocity and visibility maps over time, both indexed by initial-frame points. A semantic physical prior provides scene context. The velocities are integrated into trajectories, which guide the video generator together with the visibility maps.

2 Related Work

2.1 Controllable Video Generation

Controllable video generation aims to steer video synthesis with semantic, camera, or spatial conditions, including camera trajectories (Wang et al., 2025c; Ren et al., 2025), human poses (Zhu et al., 2024; Xu et al., 2024; Pang et al., 2025), and sketches (Zhang et al., 2021; Liu et al., 2025). Trajectory-based methods (Niu et al., 2024; Chu et al., 2025; Wan et al., 2025; Shin et al., 2025) are particularly relevant to our setting. They allow users to specify object motion by drawing trajectories, rendering moving Gaussian blobs, or warping features from the initial frame $\boldsymbol{I}_0$. These methods provide effective control over visual motion, but they typically require the user to specify the desired trajectory. In physically complex scenes, manually specifying pixel-space trajectories that obey force-driven dynamics is difficult. Our work instead predicts object trajectories from force and initial-velocity controls, rather than assuming physically valid trajectories as input.

2.2 Physics-Based Video Generation

Integrating physical simulators into video generation can introduce explicit physical priors to improve physical fidelity (Genesis Embodied AI, 2024). For example, PhysDreamer (Zhang et al., 2024) and PhysGaussian (Xie et al., 2024) combine 3D Gaussian representations with physical simulation to drive dynamic video generation. Other hybrid approaches, such as PerpetualWonder (Zhan et al., 2026) and WonderPlay (Li et al., 2025), use simulators to generate coarse dynamic guidance, which is then refined by video generation models. PhysCtrl (Wang et al., 2025a) reconstructs 3D object points and generates their trajectories under physical-parameter and force conditions. FoMo directly predicts initial-frame-indexed 2D velocity and visibility maps from an image and relative controls, then integrates velocity for video synthesis. This representation supports training from ordinary videos without reconstructed 3D geometry, metric material parameters, or 3D motion supervision.

2.3 Force-Conditioned Video Generation

Another line of work learns force-conditioned physical responses from paired force-video data. Force Prompting (Gillman et al., 2025) and Goal Force (Gillman et al., 2026) encode sparse force controls as conditioning signals for video diffusion models, lowering the barrier for force-conditioned video generation. They learn the response through an RGB video objective without an explicit intermediate motion target. FoMo instead predicts motion direction and magnitude in velocity space before video synthesis. We supervise this motion target with relative controls derived from motion changes in real videos, thereby covering more objects, materials, scenes, and interactions. The control signals in all three methods are relative rather than calibrated 3D forces.

3 METHOD

3.1 PROBLEM FORMULATION

Our goal is to model how a physical scene evolves under force and initial-velocity controls. As shown in Fig. 2, given an initial frame $\boldsymbol{I}_0$, force controls $\boldsymbol{F}$, initial velocity controls $\boldsymbol{v}$, and a semantic physical prior y, FoMo generates a video sequence $\{\boldsymbol{I}_i\}_{i=1}^n$. The force and initial-velocity controls are represented as sparse 2D maps: only the user-specified control points contain non-zero vectors, while all other locations are set to zero. The semantic physical prior y is a high-level text prompt that describes scene properties and possible motion trends under the given controls. The generated video is expected to preserve the visual appearance of the initial frame while reflecting the motion induced by the input controls. Because the controls are defined in image coordinates, their directions and magnitudes are relative values rather than measurements in Newtons or 3D world coordinates. For brevity, we refer to this relative 2D signal as a force control.

Most existing force-conditioned video generation methods (Gillman et al., 2025; 2026) learn a direct mapping from control signals to video frames,

$$(\boldsymbol{I}_0, \boldsymbol{F}, y) \rightarrow \{\boldsymbol{I}_i\}_{i=1}^n. \tag{1}$$

In contrast, FoMo decomposes the problem into motion-state prediction and video synthesis. Instead of predicting pixels directly, we first predict a motion state $\mathcal{S}$ consisting of dense object velocity and visibility maps, integrate the velocity maps into object trajectories, and then use the trajectories and visibility maps to guide video synthesis, as shown in Fig. 2:

$$(\boldsymbol{I}_0, \boldsymbol{F}, \boldsymbol{v}, y) \rightarrow \mathcal{S} \rightarrow \{\boldsymbol{X}_i\}_{i=0}^n \rightarrow \{\boldsymbol{I}_i\}_{i=1}^n. \tag{2}$$

Here, the motion state $\mathcal{S} = (\{\boldsymbol{V}_i\}_{i=0}^{n-1}, \{\boldsymbol{M}_i\}_{i=1}^n)$, where $\boldsymbol{V}_i$ and $\boldsymbol{M}_i$ denote the predicted velocity and visibility maps, respectively, and $\{\boldsymbol{X}_i\}_{i=0}^n$ denotes the integrated object trajectory maps.

Both components of the motion state $\mathcal{S}$ are indexed by initial-frame points. Specifically, for a point p initially located at $\boldsymbol{x}_0$ in $\boldsymbol{I}_0$, let $\boldsymbol{x}_i$ denote its position at frame i. The velocity $\boldsymbol{V}_i(p)$ describes its displacement from $\boldsymbol{x}_i$ to $\boldsymbol{x}_{i+1}$, while $\boldsymbol{M}_i(p)$ indicates whether the same point is visible at frame i. This initial-frame indexing differs from optical flow, which represents motion between consecutive frames.

The velocity component represents point motion, the visibility component indicates whether each point remains visible, and the pretrained generator handles appearance. Thus, "motion state" denotes the joint velocity and visibility maps. These outputs do not encode mass, momentum, contact geometry, or metric depth.

3.2 MOTION STATE PREDICTION

We predict the motion state $\mathcal{S}$ from the input controls using a diffusion network $\boldsymbol{\Phi}_V$. We initialize $\boldsymbol{\Phi}_V$ from a pretrained image-to-video diffusion model (Wan et al., 2025) and add input and output adapters for the velocity and visibility maps.

Motion State Formulation. We define the motion state as $\mathcal{S} = (\{\boldsymbol{V}_i\}_{i=0}^{n-1}, \{\boldsymbol{M}_i\}_{i=1}^n)$, a sequence of dense velocity and visibility maps indexed by points in the initial frame. To construct training targets, we first use CoTracker3 (Karaev et al., 2025) to extract object trajectory maps $\{\boldsymbol{X}_i\}_{i=0}^n$. For a tracked point p located at $\boldsymbol{x}_0$ in the initial frame $\boldsymbol{I}_0$, $\boldsymbol{X}_i(p) = \boldsymbol{x}_i$ denotes the location of the same point at frame i. We derive its velocity by taking the displacement between consecutive frames:

$$\boldsymbol{V}_i = \boldsymbol{X}_{i+1} - \boldsymbol{X}_i. \tag{3}$$

This representation removes appearance from the prediction target, allowing different motions to remain distinguishable even when the corresponding frames look similar. The tracked velocities also make motion changes observable in ordinary videos, allowing us to construct relative controls without force sensors.

Relative Motion-Change Controls from Real Videos. We begin each clip at its first frame with observable motion and derive the initial velocity and relative motion-change control from the first three tracked positions:

$$\boldsymbol{v}^{\mathrm{rel}} = \boldsymbol{X}_1 - \boldsymbol{X}_0, \qquad \boldsymbol{F}^{\mathrm{rel}} := \boldsymbol{X}_2 - 2\boldsymbol{X}_1 + \boldsymbol{X}_0. \tag{4}$$

After retaining valid tracked points, we group them by object mask and select the point with the largest initial motion change for each object. Each selected object yields one training pair. We place the initial-velocity and relative motion-change vectors at the selected initial coordinate as sparse conditions and provide them only at the first time step. Resizing, cropping, and signed-log transforms are applied consistently to the vector conditions and motion targets. Although the second-order difference is related to net force for a fixed mass, monocular video does not provide metric scale, depth, camera motion, or hidden contacts. Therefore, $\boldsymbol{F}^{\mathrm{rel}}$ represents only the direction and relative magnitude of the observed initial motion change, not an exact 3D force.

Semantic Physical Prior. Exact physical parameters, such as mass, friction coefficients, and Young's modulus, are often difficult to obtain in practical scenarios. Some physics-based video generation methods, such as PhysCtrl (Wang et al., 2025a), rely on such parameters to model physical interactions, while force-conditioned video generation methods (Gillman et al., 2025; 2026) typically use text prompts that describe the intended motion without explicitly grounding them in the input controls. In contrast, we use a text prompt y as a semantic physical prior that provides high-level physical context without requiring exact simulator parameters.

Given the initial frame $\boldsymbol{I}_0$, sparse force controls $\boldsymbol{F}$, and initial-velocity controls $\boldsymbol{v}$, we first visualize the controls as colored arrows at their interaction points. A vision-language model then uses this marked frame to generate a prompt describing object properties, scene conditions, and possible motion trends. Unlike prompts generated without visualized controls (Gillman et al., 2025; 2026), this prompt explicitly includes the specified force and initial velocity and is used to condition both motion-state prediction and trajectory-conditioned video synthesis.

Diffusion for Motion State Prediction. We formulate motion-state generation as a conditional denoising process. Given the initial frame $\boldsymbol{I}_0$, text prompt y, and sparse controls, the model predicts the joint velocity and visibility representation $\mathcal{S}$. We adapt a DiT-based architecture (Wan et al., 2025) to operate directly in motion space. Specifically, we concatenate the sparse force $\boldsymbol{F}$ and initial velocity $\boldsymbol{v}$ with the encoded features of $\boldsymbol{I}_0$ along the channel dimension. Both controls are spatially aligned maps in which only the interaction points contain non-zero vectors. Unlike standard video diffusion, we do not temporally compress the velocity sequence, thereby preserving frame-level motion changes. We add linear input and output adapters to match the pretrained model to the joint representation.

Visibility Prediction. As part of the motion state, the network predicts visibility maps $\{\boldsymbol{M}_i\}_{i=1}^{n}$ for the initial-frame points. Each map indicates whether the corresponding point remains visible at frame i, which allows the downstream trajectory-conditioned generator to ignore invalid points. We obtain pseudo ground-truth visibility labels from CoTracker (Karaev et al., 2025) and train the visibility and velocity heads jointly.

3.3 Motion-Space Diffusion Objective

We train the motion state prediction network $\boldsymbol{\Phi}_V$ with a v-prediction objective (Salimans & Ho, 2022; Wan et al., 2025). We adapt the objective to the large numerical range of velocities and the spatial sparsity of force-induced motion.

Log-space velocity mapping. Velocity maps may have a much larger dynamic range than normalized video latents. To stabilize training, we transform velocities into a signed logarithmic space and concatenate them with the visibility maps as the diffusion target:

$$\begin{aligned} \tilde{\boldsymbol{V}}_i &= \mathrm{sgn}(\boldsymbol{V}_i)\log(1+|\boldsymbol{V}_i|), \\ \boldsymbol{Y} &= [\{\tilde{\boldsymbol{V}}_i\}_{i=0}^{n-1}, \{\boldsymbol{M}_i\}_{i=1}^{n}]. \end{aligned} \tag{5}$$

This signed log-space transformation is monotonic and invertible for each velocity component, preserving the sign while compressing large magnitudes. The visibility maps are optimized jointly with the velocity maps under the same diffusion objective.

Motion-balanced diffusion loss. Since moving objects usually occupy only a small portion of the image, a uniform diffusion loss can be dominated by static regions and bias the model toward near-zero velocities. During training, we define a pointwise motion mask from ground-truth trajectories:

$$\boldsymbol{M}_{\mathrm{filt}}(p) = \mathbf{1}[d_{\mathrm{disp}}(p) \geq \tau_{\mathrm{disp}} \ \vee \ d_{\mathrm{step}}(p) \geq \tau_{\mathrm{step}}], \tag{6}$$

The displacement terms are defined in Appendix G.2. We use $\tau_{\text{disp}} = 10$ and $\tau_{\text{step}} = 5$ pixels on the 256×256 grid. We use this mask to average the diffusion loss $\boldsymbol{E}_\tau$ separately over moving and relatively static regions:

$$\mathcal{L}_{\text{mov}} = \frac{\sum \boldsymbol{M}_{\text{filt}} \odot \boldsymbol{E}_\tau}{\sum \boldsymbol{M}_{\text{filt}} + \epsilon}, \qquad \mathcal{L}_{\text{stat}} = \frac{\sum (1 - \boldsymbol{M}_{\text{filt}}) \odot \boldsymbol{E}_\tau}{\sum (1 - \boldsymbol{M}_{\text{filt}}) + \epsilon}, \tag{7}$$
$$\mathcal{L} = \mathcal{L}_{\text{mov}} + \mathcal{L}_{\text{stat}} + \lambda_\Delta \mathcal{L}_\Delta + \lambda_{\text{end}} \mathcal{L}_{\text{end}}.$$

Here, ϵ is a small constant for numerical stability. Normalizing the two regions separately prevents static points from overwhelming force-induced motion and stabilizes motion-state prediction.

Motion Regularization Terms. The term $\mathcal{L}_\Delta$ emphasizes spatiotemporal locations with abrupt ground-truth velocity changes, which often correspond to contact or rapid acceleration. After initial diffusion training stabilizes, $\mathcal{L}_{\text{end}}$ applies one-step denoising, integrates the recovered velocities into trajectories, and supervises the final pixel offset to reduce accumulated integration error. Detailed definitions of both terms are provided in the supplementary material.

3.4 Trajectory-conditioned Video Synthesis

Given the predicted motion state $\mathcal{S}$, we reconstruct dense object trajectories from its velocity maps and use the visibility maps to mask invalid trajectory points.

Trajectory Reconstruction. The predicted velocity maps describe the frame-to-frame motion of points anchored in the initial frame. We integrate these velocities over time to recover trajectory maps $\{\boldsymbol{X}_i\}_{i=0}^n$. Starting from the initial coordinates $\boldsymbol{X}_0$, the trajectory at frame i is computed as

$$\boldsymbol{X}_i = \boldsymbol{X}_0 + \sum_{j=0}^{i-1} \boldsymbol{V}_j. \tag{8}$$

The resulting dense trajectories provide explicit spatial guidance for the subsequent video synthesis stage.

Motion-conditioned Trajectory Filtering. Small residual velocities can arise from diffusion errors and tracker drift (Karaev et al., 2025). After temporal integration, they can create false motion in stationary regions. At inference, we apply the mask in Eq. 6 to the predicted trajectories and set $\boldsymbol{X}_i(p) = \boldsymbol{X}_0(p)$ for all i when $\boldsymbol{M}_{\text{filt}}(p) = 0$, before video synthesis.

Trajectory-Conditioned Video Rendering. We use a pretrained trajectory-conditioned video generator to synthesize the final frames. Existing trajectory-conditioned models (Chu et al., 2025; Jiang et al., 2025) are primarily trained for human-centric motion and may not follow object-centric trajectories accurately. We therefore apply orthogonal fine-tuning (Qiu et al., 2025) with ground-truth object trajectories to adapt the backbone to this setting. The generator takes the initial frame $\boldsymbol{I}_0$, text prompt y, and integrated trajectory maps $\{\boldsymbol{X}_i\}_{i=1}^n$ as conditioning inputs, together with the visibility maps $\{\boldsymbol{M}_i\}_{i=1}^n$ from the predicted motion state. The visibility maps act as validity masks, allowing the renderer to ignore occluded or unreliable trajectory points.

3.5 Inference and Rollout

Fixed-horizon inference. At inference, the motion prediction network first estimates dense velocity maps $\{\boldsymbol{V}_i\}_{i=0}^{n-1}$ and visibility maps $\{\boldsymbol{M}_i\}_{i=1}^n$ from the initial frame $\boldsymbol{I}_0$, sparse force controls $\boldsymbol{F}$, initial-velocity controls $\boldsymbol{v}$, and semantic physical prior y. The velocity maps are then integrated into trajectory maps $\{\boldsymbol{X}_i\}_{i=0}^n$ following Eq. 8, and the resulting trajectories and visibility maps are passed to the trajectory-conditioned video generator described in Section 3.4.

Motion-state rollout. This interface supports overlapping rollout beyond one segment. The next segment begins at the third-to-last frame, so the last three predicted positions define inherited velocity and motion-change controls through the first- and second-order differences in Eq. 4. The user may replace the inherited force control and update the semantic prior. Repeating this process supports longer generation and multi-round interaction, although motion and appearance errors can accumulate.

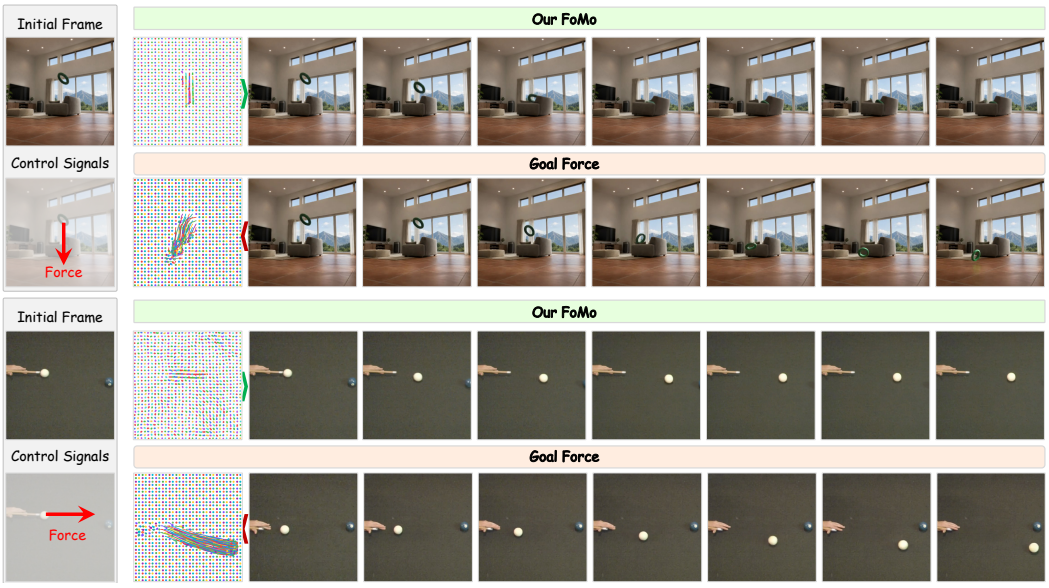

Figure 3: Selected comparisons with Goal Force (Gillman et al., 2026). CoTracker3 (Karaev et al., 2025) trajectories show jitter in the top example and motion that does not follow the control in the bottom example. FoMo produces smoother 2D motion that follows the control more closely.

Table 1: Pairwise 2AFC preference for FoMo over each baseline, based on 100 participants. Goal Force (same data) is fine-tuned on the same training dataset as FoMo.

Comparison	Contact/Collision			Drop/Rebound			Slide/Roll/Throw			Multi-object		
	Force (%)	Phys. (%)	Visual (%)	Force (%)	Phys. (%)	Visual (%)	Force (%)	Phys. (%)	Visual (%)	Force (%)	Phys. (%)	Visual (%)
Wan2.2-A14B-I2V	90.94	81.46	88.07	90.06	85.73	82.98	92.00	80.30	84.40	98.70	98.95	98.30
Goal Force	80.46	80.73	81.68	75.36	72.20	66.70	82.00	79.10	78.95	98.80	97.80	98.10
Goal Force (same data)	72.51	76.83	75.32	70.32	73.31	67.52	80.92	75.37	72.22	92.77	94.52	90.88

4 Experiments

4.1 Implementation Details

Datasets We combine Physion videos (Bear et al., 2021; Tung et al., 2023) with real-world data from Pexels[1], Physics101 (Wu et al., 2016), and OpenVid (Nan et al., 2025). Filtering and slicing about 18,000 Physion videos yields about 100,000 synthetic clips. Selecting static-camera videos without human-centered motion, followed by frame extraction, motion filtering, cropping, and resizing, yields about 200,000 real clips. We use separate sets for evaluation: a 200-sample set supports motion metrics, while the 80-sample benchmark below supports the human study. The motion set uses the same initial-frame tracking representation as training. The supplement provides data and reproducibility details. We will release the fixed data lists, prompts, controls, model configurations, and evaluation scripts.

Network Implementation We initialize the motion-state predictor from Wan2.1-Fun-1.3B-InP and add randomly initialized linear input and output adapters. Qwen3-VL-235B-A22B-Instruct generates the semantic physical prior. Each 25-frame clip includes the initial frame, so the predictor produces 24 velocity maps and 24 visibility maps at 256×256 resolution on a 32×32 initial-frame point grid with an 8-pixel stride, using 50 denoising steps at inference. Finally, Wan-Move (Chu et al., 2025) uses the integrated trajectories to render the corresponding 25-frame video at 480×480 resolution with 40 sampling steps.

[1] https://www.pexels.com/

Table 2: Core design ablations. Lower motion error and higher interaction success are better.

Setting	Vel. ↓	Traj. ↓	Succ. ↑
Acc. target	0.237	1.711	91.22%
Traj. target	0.562	2.317	87.31%
w/o Sem. prior	0.229	1.787	82.57%
w/o Real data	0.862	2.217	69.80%
Full FoMo	**0.211**	**1.662**	**97.31%**

Table 3: Motion regularization on Physion; lower is better.

	Vel. Err.↓		Traj. Err.↓	
Setting	Moving	Static	Moving	Static
w/o $\boldsymbol{M}_{\text{filt}}$	1.027	**0.011**	4.987	**0.101**
w/o $\mathcal{L}_{\Delta}$	**0.207**	0.016	1.712	0.133
w/o $\mathcal{L}_{\text{end}}$	0.210	0.021	1.921	0.141
Full FoMo	0.211	0.017	**1.662**	0.137

Training Configuration We train the motion-state predictor on eight A100-80GB GPUs with a total batch size of 16. We use full supervised fine-tuning and AdamW (Loshchilov & Hutter, 2017) with a learning rate of 10^{-4} for 200K steps.

4.2 Comparisons with Baseline Methods

We compare the proposed FoMo with Goal Force (Gillman et al., 2026), a recent force-conditioned video generation baseline, through visual comparisons and pairwise 2AFC human evaluations. Following the recommended Goal Force setting, we set the indirect-force and mass channels to 0 and -1, respectively. In all comparisons, Goal Force receives the same control-aware semantic prompt as FoMo. We additionally fine-tune it on FoMo's training data, convert each relative motion-change control to its moving-Gaussian representation, and match the output length, resolution, and sampling steps. The native interfaces differ in initial velocity: FoMo receives a sparse vector map, while Goal Force receives the same information through the prompt. This matched-data comparison evaluates complete systems with native architectures and control interfaces. Prediction-target ablations evaluate representation choices within the same FoMo backbone. We also include Wan2.2-A14B-I2V (Wan et al., 2025), the image-to-video backbone used by Goal Force, as a text-only reference. It receives the same initial frame and control-aware prompt because it does not accept vector controls.

The training set and both evaluation sets are source-video-disjoint, with no overlapping frames or clips. Our 80-sample benchmark contains 60 initial frames from openly licensed sources, including Pexels and OpenVid (Nan et al., 2025), and 20 generated by GPT Image-2. It covers contact and collision, drops and rebounds, sliding, rolling and throwing, and multi-object interactions. Because different generation systems can produce different valid futures, reference-based pixel metrics can mainly reflect appearance rather than control consistency. We therefore use pairwise 2AFC evaluation. Each trial shows the input control and two anonymized videos generated from the same input. Method order is randomized; questionnaire shuffling and length adjustments are used to reduce participant fatigue. The 100 participants may replay videos and judge force consistency, physical plausibility, and visual quality separately.

Table 1 shows that the preference for FoMo exceeds 50% for every baseline, scenario, and criterion. Against Goal Force trained on the same data, preference rates range from 67.52% to 94.52%, with the highest rates for multi-object interactions. The matched-data comparison evaluates complete pipelines; the target ablations isolate representation choices within FoMo. For comparisons with Wan2.2 and the original Goal Force, all rates remain significant under one-sided binomial tests with Bonferroni correction. The lowest rate among these comparisons, 66.70%, gives $p = 4.37 \times 10^{-4}$ under a conservative participant-level test. These results measure visible control consistency and human judgment; physical plausibility is not a test of exact physical laws.

4.3 Core Design Ablations

Table 2 evaluates the main design choices on the 200-sample motion set. The first two rows replace the velocity target with acceleration or trajectory, while the next two remove the semantic prior or real-video training. All rows use the same FoMo backbone and protocol. Velocity and trajectory errors are ℓ_1 differences from CoTracker-derived velocity and integrated position targets over moving regions on the 256×256 initial-frame grid. Interaction success requires the controlled object to follow the control and produce the intended visible event; missing responses and unintended motion are failures. This metric does not measure force or energy. Acceleration represents motion change directly but requires an additional temporal difference, which makes its target more sensitive to tracking noise near contact. Velocity avoids this extra difference and gives lower errors and higher

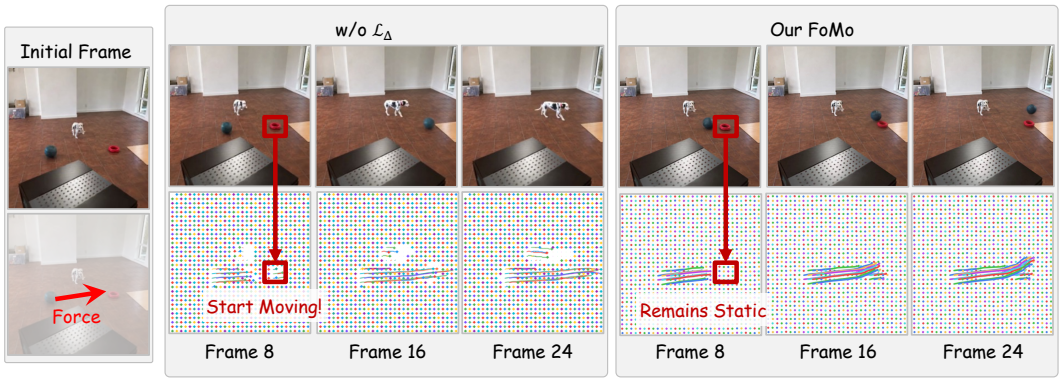

Figure 4: Effect of abrupt-change supervision. Without $\mathcal{L}_\Delta$, the object starts moving before contact, whereas the full model keeps it static until contact.

success. Direct trajectory prediction raises trajectory error by 39.4%, supporting the value of supervising local motion before integration. Removing the semantic prior changes motion errors only modestly but lowers success to 82.57%, indicating that it mainly supplies scene context. Removing real videos raises velocity error from 0.211 to 0.862 and lowers success to 69.80%, showing the importance of broader training coverage.

4.4 Motion Regularization Ablations

Table 3 evaluates the three motion regularizers on Physion under identical conditions. Removing M_{filt}, which balances moving and static regions, raises moving-region velocity error from 0.211 to 1.027 and trajectory error from 1.662 to 4.987. Removing $\mathcal{L}_\Delta$ changes aggregate errors only slightly, but Fig. 4 shows premature motion before contact. Removing $\mathcal{L}_{\text{end}}$ leaves local velocity error nearly unchanged but raises trajectory error from 1.662 to 1.921. The three terms therefore address spatial imbalance, event timing, and accumulated integration error, respectively.

5 Limitations

Our method depends on reliable motion estimation and trajectory-conditioned rendering. Blur, occlusion, tracking errors, and rendering artifacts can reduce performance, especially for small, briefly hidden, or fast-moving objects. Even an accurate 2D trajectory leaves rotation, depth, deformation, and contact geometry to the renderer, which may produce a visually plausible but physically incorrect response. The inputs are also approximate: the semantic prior provides context rather than calibrated properties, and the motion change used to construct a relative control cannot reveal whether applied force, gravity, friction, hidden contact, camera motion, or depth caused the response. Errors can accumulate during rollout because the last three predicted positions initialize the overlapping next segment. Finally, our evaluation measures visible motion and human preference rather than conservation laws, hidden forces, or a unique physical future. We therefore claim relative 2D control and plausible visible responses, not metric force recovery or exact physics.

6 Conclusion

We presented FoMo, which predicts initial-frame-indexed velocity and visibility maps, integrates the velocities into trajectories, and synthesizes force-conditioned video. This separation provides direct motion supervision while retaining a pretrained generator's visual prior. Relative controls from real videos broaden the training objects and interactions, while semantic physical priors add context without exact simulator parameters. The explicit motion state also supports overlapping rollout with updated controls, although errors can accumulate. On the tested benchmark, participants prefer FoMo over text-only Wan2.2, the original Goal Force, and Goal Force fine-tuned on the same data. The same-data comparison reduces training-data differences, and controlled ablations support velocity prediction and real-video training.

AI USE STATEMENT

GPT Image-2 was used to generate 20 initial frames in the human-evaluation benchmark, as reported above. OpenAI Codex assisted with language editing, consistency and compliance checks, and LaTeX/PDF formatting and verification; it was not used to generate experiments or results. The authors reviewed and verified all technical statements, citations, equations, experimental details, and reported results and remain responsible for the submitted material.

REFERENCES

Niket Agarwal, Arslan Ali, Maciej Bala, Yogesh Balaji, Erik Barker, Tiffany Cai, Prithvijit Chattopadhyay, Yongxin Chen, Yin Cui, Yifan Ding, et al. Cosmos world foundation model platform for physical ai. *arXiv preprint arXiv:2501.03575*, 2025.

Daniel M Bear, Elias Wang, Damian Mrowca, Felix J Binder, Hsiao-Yu Fish Tung, RT Pramod, Cameron Holdaway, Sirui Tao, Kevin Smith, Fan-Yun Sun, et al. Physion: Evaluating physical prediction from vision in humans and machines. *arXiv preprint arXiv:2106.08261*, 2021.

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.

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 *IEEE Conf. Comput. Vis. Pattern Recog.*, pp. 13–23, 2025.

Ernie Chu, Shuo-Yen Lin, and Jun-Cheng Chen. Video controlnet: Towards temporally consistent synthetic-to-real video translation using conditional image diffusion models. *arXiv preprint arXiv:2305.19193*, 2023.

Ruihang Chu, Yefei He, Zhekai Chen, Shiwei Zhang, Xiaogang Xu, Bin Xia, Dingdong WANG, Hongwei Yi, Xihui Liu, Hengshuang Zhao, et al. Wan-move: Motion-controllable video generation via latent trajectory guidance. In *Adv. Neural Inform. Process. Syst.*, 2025.

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

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. *Adv. Neural Inform. Process. Syst.*, 2025.

Nate Gillman, Yinghua Zhou, Zitian Tang, Evan Luo, Arjan Chakravarthy, Daksh Aggarwal, Michael Freeman, Charles Herrmann, and Chen Sun. Goal force: Teaching video models to accomplish physics-conditioned goals. *IEEE Conf. Comput. Vis. Pattern Recog.*, 2026.

Yuwei Guo, Ceyuan Yang, Anyi Rao, Zhengyang Liang, Yaohui Wang, Yu Qiao, Maneesh Agrawala, Dahua Lin, and Bo Dai. Animatediff: Animate your personalized text-to-image diffusion models without specific tuning. *arXiv preprint arXiv:2307.04725*, 2023.

Jonathan Ho, William Chan, Chitwan Saharia, Jay Whang, Ruiqi Gao, Alexey Gritsenko, Diederik P Kingma, Ben Poole, Mohammad Norouzi, David J Fleet, et al. Imagen video: High definition video generation with diffusion models. *arXiv preprint arXiv:2210.02303*, 2022.

Zeyinzi Jiang, Zhen Han, Chaojie Mao, Jingfeng Zhang, Yulin Pan, and Yu Liu. Vace: All-in-one video creation and editing. In *Int. Conf. Comput. Vis.*, pp. 17191–17202, 2025.

Nikita Karaev, Yuri Makarov, Jianyuan Wang, Natalia Neverova, Andrea Vedaldi, and Christian Rupprecht. Cotracker3: Simpler and better point tracking by pseudo-labelling real videos. In *Int. Conf. Comput. Vis.*, pp. 6013–6022, 2025.

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 *Int. Conf. Comput. Vis.*, pp. 9080–9090, 2025.

Feng-Lin Liu, Hongbo Fu, Xintao Wang, Weicai Ye, Pengfei Wan, Di Zhang, and Lin Gao. Sketchvideo: Sketch-based video generation and editing. In *IEEE Conf. Comput. Vis. Pattern Recog.*, pp. 23379–23390, 2025.

Ilya Loshchilov and Frank Hutter. Decoupled weight decay regularization. *arXiv preprint arXiv:1711.05101*, 2017.

Yongrui Ma, Shijie Zhao, Mingde Yao, Junlin Li, Xiaohong Liu, Qi Dou, Jinwei Gu, Tianfan Xue, et al. Realtime video frame interpolation using one-step diffusion sampling. In *Int. Conf. Learn. Represent.*, 2026.

Henrique Morimitsu, Xiaobin Zhu, Roberto M Cesar, Xiangyang Ji, and Xu-Cheng Yin. Dpflow: Adaptive optical flow estimation with a dual-pyramid framework. In *IEEE Conf. Comput. Vis. Pattern Recog.*, pp. 17810–17820, 2025.

Kepan Nan, Rui Xie, Penghao Zhou, Tiehan Fan, Zhenheng Yang, Zhijie Chen, Xiang Li, Jian Yang, and Ying Tai. Openvid-1m: A large-scale high-quality dataset for text-to-video generation. In *Int. Conf. Learn. Represent.*, volume 2025, pp. 1045–1064, 2025.

Muyao Niu, Xiaodong Cun, Xintao Wang, Yong Zhang, Ying Shan, and Yinqiang Zheng. Mofa-video: Controllable image animation via generative motion field adaptions in frozen image-to-video diffusion model. In *Eur. Conf. Comput. Vis.*, pp. 111–128. Springer, 2024.

Yatian Pang, Bin Zhu, Bin Lin, Mingzhe Zheng, Francis EH Tay, Ser-Nam Lim, Harry Yang, and Li Yuan. Dreamdance: Animating human images by enriching 3d geometry cues from 2d poses. In *Int. Conf. Comput. Vis.*, pp. 14039–14050, 2025.

Zeju Qiu, Weiyang Liu, Adrian Weller, and Bernhard Schölkopf. Orthogonal finetuning made scalable. In *Conf. Empirical Methods Natural Language Process.*, pp. 31934–31951, 2025.

Xuanchi Ren, Tianchang Shen, Jiahui Huang, Huan Ling, Yifan Lu, Merlin Nimier-David, Thomas Müller, Alexander Keller, Sanja Fidler, and Jun Gao. Gen3c: 3d-informed world-consistent video generation with precise camera control. In *IEEE Conf. Comput. Vis. Pattern Recog.*, 2025.

Tim Salimans and Jonathan Ho. Progressive distillation for fast sampling of diffusion models. *arXiv preprint arXiv:2202.00512*, 2022.

Xiaoyu Shi, Zhaoyang Huang, Fu-Yun Wang, Weikang Bian, Dasong Li, Yi Zhang, Manyuan Zhang, Ka Chun Cheung, Simon See, Hongwei Qin, et al. Motion-i2v: Consistent and controllable image-to-video generation with explicit motion modeling. In *Proc. ACM SIGGRAPH*, pp. 1–11, 2024.

Joonghyuk Shin, Zhengqi Li, Richard Zhang, Jun-Yan Zhu, Jaesik Park, Eli Shechtman, and Xun Huang. Motionstream: Real-time video generation with interactive motion controls. *arXiv preprint arXiv:2511.01266*, 2025.

Robbyant Team, Zelin Gao, Qiuyu Wang, Yanhong Zeng, Jiapeng Zhu, Ka Leong Cheng, Yixuan Li, Hanlin Wang, Yinghao Xu, Shuailei Ma, Yihang Chen, Jie Liu, Yansong Cheng, Yao Yao, Jiayi Zhu, Yihao Meng, Kecheng Zheng, Qingyan Bai, Jingye Chen, Zehong Shen, Yue Yu, Xing Zhu, Yujun Shen, and Hao Ouyang. Advancing open-source world models. *arXiv preprint arXiv:2601.20540*, 2026.

Zachary Teed and Jia Deng. Raft: Recurrent all-pairs field transforms for optical flow. In *Eur. Conf. Comput. Vis.*, pp. 402–419. Springer, 2020.

Hsiao-Yu Tung, Mingyu Ding, Zhenfang Chen, Daniel Bear, Chuang Gan, Josh Tenenbaum, Dan Yamins, Judith Fan, and Kevin Smith. Physion++: Evaluating physical scene understanding that requires online inference of different physical properties. *Advances in Neural Information Processing Systems*, 36:67048–67068, 2023.

Team Wan, Ang Wang, Baole Ai, Bin Wen, Chaojie Mao, Chen-Wei Xie, Di Chen, Feiwu Yu, Haiming Zhao, Jianxiao Yang, Jianyuan Zeng, et al. Wan: Open and advanced large-scale video generative models. *arXiv preprint arXiv:2503.20314*, 2025.

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. *arXiv preprint arXiv:2509.20358*, 2025a.

Jiaming Wang, Diwen Liu, Jizhuo Chen, Jiaxuan Da, Nuowen Qian, Minh Man Tram, and Harold Soh. Genie: A generalizable navigation system for in-the-wild environments. *IEEE Robotics and Automation Letters*, 2025b.

Qinghe Wang, Yawen Luo, Xiaoyu Shi, Xu Jia, Huchuan Lu, Tianfan Xue, Xintao Wang, Pengfei Wan, Di Zhang, and Kun Gai. Cinemaster: A 3d-aware and controllable framework for cinematic text-to-video generation. In *Proc. ACM SIGGRAPH*, pp. 1–10, 2025c.

Jiajun Wu, Joseph J Lim, Hongyi Zhang, Joshua B Tenenbaum, and William T Freeman. Physics 101: Learning physical object properties from unlabeled videos. In *BMVC*, pp. 7, 2016.

Tianyi Xie, Zeshun Zong, Yuxing Qiu, Xuan Li, Yutao Feng, Yin Yang, and Chenfanfu Jiang. Physgaussian: Physics-integrated 3d gaussians for generative dynamics. In *IEEE Conf. Comput. Vis. Pattern Recog.*, pp. 4389–4398, 2024.

Zhongcong Xu, Jianfeng Zhang, Jun Hao Liew, Hanshu Yan, Jia-Wei Liu, Chenxu Zhang, Jiashi Feng, and Mike Zheng Shou. Magicanimate: Temporally consistent human image animation using diffusion model. In *IEEE Conf. Comput. Vis. Pattern Recog.*, pp. 1481–1490, 2024.

Jiahao Zhan, Zizhang Li, Hong-Xing Yu, and Jiajun Wu. Perpetualwonder: Long-horizon action-conditioned 4d scene generation. *arXiv preprint arXiv:2602.04876*, 2026.

Haichao Zhang, Gang Yu, Tao Chen, and Guozhong Luo. Sketch me a video. *arXiv preprint arXiv:2110.04710*, 2021.

Lvmin Zhang, Anyi Rao, and Maneesh Agrawala. Adding conditional control to text-to-image diffusion models. In *Int. Conf. Comput. Vis.*, pp. 3836–3847, 2023.

Tianyuan Zhang, Hong-Xing Yu, Rundi Wu, Brandon Y. Feng, Changxi Zheng, Noah Snavely, Jiajun Wu, and William T. Freeman. Physdreamer: Physics-based interaction with 3d objects via video generation. In *Eur. Conf. Comput. Vis.*, 2024.

Shenhao Zhu, Junming Leo Chen, Zuozhuo Dai, Zilong Dong, Yinghui Xu, Xun Cao, Yao Yao, Hao Zhu, and Siyu Zhu. Champ: Controllable and consistent human image animation with 3d parametric guidance. In *Eur. Conf. Comput. Vis.*, pp. 145–162. Springer, 2024.