PhysGen: Rigid-Body Physics-Grounded Image-to-Video Generation

Shaowei Liu, Zhongzheng Ren, Saurabh Gupta*, and Shenlong Wang*

University of Illinois Urbana-Champaign
https://stevenlsw.github.io/physgen/

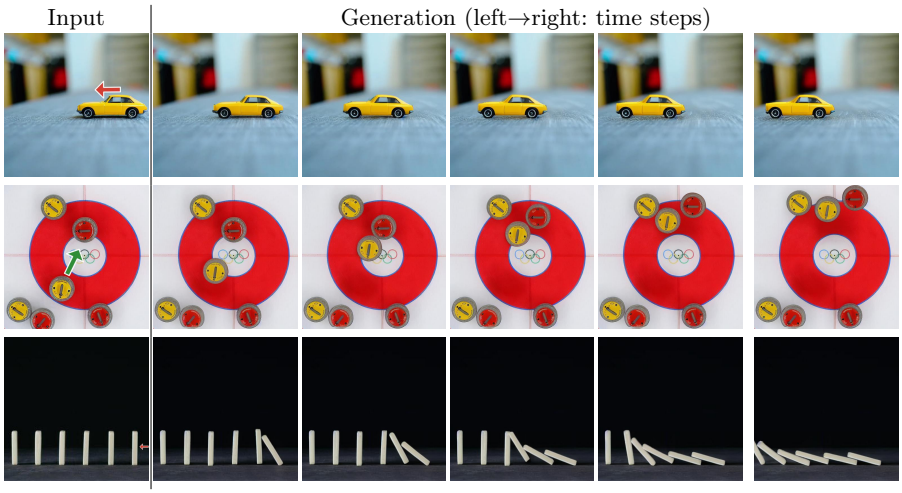

Fig. 1: Given a single image, training-free PhysGen generates future frames controlled by physics and initial state. Semantics, geometry and dynamics are reasoned during generation and the result videos are both physics-grounded and photo-realistic.

Abstract. We present PhysGen, a novel image-to-video generation method that converts a single image and an input condition (*e.g.*, force and torque applied to an object in the image) to produce a realistic, physically plausible, and temporally consistent video. Our key insight is to integrate model-based physical simulation with a data-driven video generation process, enabling plausible image-space dynamics. At the heart of our system are three core components: (i) an image understanding module that effectively captures the geometry, materials, and physical parameters of the image; (ii) an image-space dynamics simulation model that utilizes rigid-body physics and inferred parameters to simulate realistic behaviors; and (iii) an image-based rendering and refinement module that leverages generative video diffusion to produce realistic video footage featuring the simulated motion. The resulting videos are realistic in both physics and appearance and are even precisely controllable, showcasing superior results over existing data-driven image-to-video generation works through quantitative comparison and comprehensive user study. PhysGen's resulting videos can be used for various downstream

* Equal advising

applications, such as turning an image into a realistic animation or allowing users to interact with the image and create various dynamics.

1 Introduction

Looking at the images in Fig. 1, we can effortlessly predict or visualize the potential outcomes of various physical effects applied to the car, the curling stone, and the stack of dominos. This understanding of physics empowers our imagination of the counterfactual: we can mentally imagine various consequences of an action from an image without experiencing them. We aim to provide computers with similar capabilities – understanding and simulating physics from a single image and creating realistic animations. We expect the resulting video to produce realistic animations with physically plausible dynamics and interactions.

Tremendous progress has been made in generating realistic and physically plausible video footage. Conventional graphics leverage model-based dynamics to simulate plausible motions and utilize a graphics renderer to simulate images. Such methods provide realistic and controllable dynamics. However, both the dynamics of physics and lighting physics are predefined, making it hard to achieve our goal of animating an image captured in the real world. On the other hand, data-driven image-to-video (I2V) generation techniques [6, 8, 58] learn to create realistic videos from a single image through training diffusion models over internet-scale data. However, despite advancements in video generative models [6, 8, 58], the incorporation of real-world physics principles into the video generation process remains largely unexplored and unsolved. Consequently, the synthesized videos often lack temporal coherence and fail to replicate authentic object motions observed in reality. Furthermore, such text-driven methods also lack fine-grained controllability. For instance, they cannot simulate the consequences of different forces and torques applied to an object.

In light of this gap, we propose a paradigm shift in video generation through the introduction of *model-based video generative models*. In contrast to existing purely generative methods, which are trained in a data-driven manner and rely on diffusion models to learn image space dynamics via scaling laws, we propose grounding object dynamics explicitly using rigid body physics, thereby integrating fundamental physical principles into the generation process. As classical mechanics theory reveals, the motion of an object is determined by its physical properties (*e.g.*, mass, elasticity, surface roughness) and external factors (*e.g.*, external forces, environmental conditions, boundary conditions).

We tackle our image-to-video generation problem via a computational framework, PhysGen. PhysGen consists of three stages: 1) image-based physics understanding; 2) model-based dynamics simulation; and 3) generative video rendering with dynamics guidance. Our method first infers object compositions and physical parameters from the input image through large visual foundation model-based reasoning (Sec. 3.1). Given an input force, we then utilize the inferred physical parameters to perform realistic dynamic simulations for each object, accounting for their rigid body dynamics, collisions, frictions, and elasticity

(Sec. 3.2). Finally, guided by motion and the input image, we present a novel generative video diffusion-based renderer that outputs the final realistic video footage (Sec. 3.3). Several illustrative examples of our framework are shown in Fig. 1. As shown in this figure, our approach combines the best of the worlds of data-driven generative modeling's appearance realism and model-based dynamics's verified physical plausibility. Notably, our generation process is highly controllable, allowing users to specify underlying physics parameters and initial conditions. This capability facilitates a range of interactive applications. Moreover, our proposed generation pipeline operates solely during inference time, eliminating the need for any training.

We evaluate PhysGen generation ability on multiple data source including the web and self-captured images. We compare against both state-of-the-art image-to-video models [19, 88, 96] as well as image editing method [29] quantitatively and qualitatively. We also perform user-study to evaluate the physical-realism and photo-realism of the generated video. Our method demonstrates physics-informed results for controllable image-to-video generation, producing realistic video sequences with grounded rigid-body dynamics and interaction. PhysGen combines learning-based generative approaches with traditional model-based physics to deliver visually appealing results without any training. Our approach harnesses a plethora of ingredients from previous works, particularly recent progress in large foundational visual models for segmentation [41, 49, 62, 94], physical understanding [2, 57], normal estimation [24, 27], relighting, and generative rendering [8, 19]. While these individual ingredients exist, our system combines them in a novel manner to enable an exciting new capacity, featuring physical plausibility for video generation at an unprecedented level.

Our contributions can be summarized as follows: 1) We propose a novel image-space dynamics model that can understand physical parameters from a single image and simulate realistic, controllable, and interactive dynamics. 2) We present a novel image-to-video system by integrating our presented dynamics model with generative diffusion-based video priors. The model is training-free, and the resulting videos are highly realistic, physically plausible, and controllable, consistently surpassing existing state-of-the-art video generative models.

2 Related work

Image-based physics reasoning and simulation. Our image-based dynamics model is built upon physical simulation. Symbolic physics simulators or physics engines have been extensively studied for various physical processes, including rigid body dynamics, fluid simulation, and deformable objects [37–39, 42, 50, 56, 73]. Such methods have been widely used across graphics, scientific discovery, robotics, and video effects. However, these rule-based or solver-based simulators face limitations in terms of expressiveness, efficiency, generalizability, and parameter tuning. To overcome these challenges, neural physics, which combines traditional physics simulators/engines with deep neural networks, has been developed to model dynamic processes such as planar pushing and bouncing [4], as

well as object interactions [3]. Domain-specific network architectures [7,46,55,64] have been developed to address various physical problems. Additionally, the integration of neural scene representations or large language models with physics simulators [44,51,87] has also been investigated recently. Nevertheless, most forward physics models, whether neural or not, have not resolved the dependency on pre-defined physical parameters, which are often specified by users.

Reasoning physical parameters from visual data allows us to simulate without manually defined physics, further enhancing convenience and realism for image-based physics simulation. To enable this, subsequent research efforts [22, 44, 45, 81–83, 85, 89] have introduced pipelines that first extract scene representations of the physical world from images or videos using neural networks, and then utilize these representations for either physics reasoning or simulation. Despite the promise of these approaches, they primarily rely on synthetic data for training, which might suffer from generalization issues, or on inverse physics, which requires inference from observed physical phenomena, often requiring video observations instead of a single image. In our work, we aim to generate physics-grounded, high-fidelity predictive videos from static input images of real-world objects with complex backgrounds. Contrary to existing work, we investigate leveraging the power of large pretrained visual foundational models [2,41,57] for physics reasoning in a zero-shot manner.

Video generative model. Video generative models have experienced remarkable advancements in recent years [6,8,12,19,28,30,32,34,43,58,66,75,77,88,93,96], with the state-of-the-art framework [58] achieving the capability to generate photo-realistic and coherent imaginative videos from text instructions using diffusion models [35, 59, 63, 67, 69]. Despite progress, challenges remain, such as the need for extensive training data, object permanence, realistic lighting, and reducing unrealistic motion and physics violations. Overcoming these is key for advancing video generative modeling and its applications, especially in physics-accurate areas like scientific discovery and robotics. Our work addresses some issues by proposing a novel image-to-video framework that merges a model-based approach with a pretrained video prior, grounding the generated content with real-world physics.

Image animation. Our method is heavily inspired by image-based animation [20, 23, 36, 65, 71, 78, 90] and cinemagraphs. One common way to improve quality is to incorporate class-specific heuristics [20,40], which can be physically simulated and integrated into the generation process. Recent advancements in deep learning have led to data-driven solutions in this domain, where temporal neural networks are trained on video datasets to directly predict consecutive video frames from an input image [10, 18, 25, 31, 36, 76]. Moreover, there is growing interest in interactive [9, 11, 47] and controllable [1, 23] image-to-video synthesis. To further improve temporal coherence, more priors such as motion fields [25,36,52,53,70,90] or flows [14,29,97], 3D geometry information [61,80], and user annotations [47] have been subsequently introduced. In this work, we propose a model-based solution and tackle the image animation task from a new

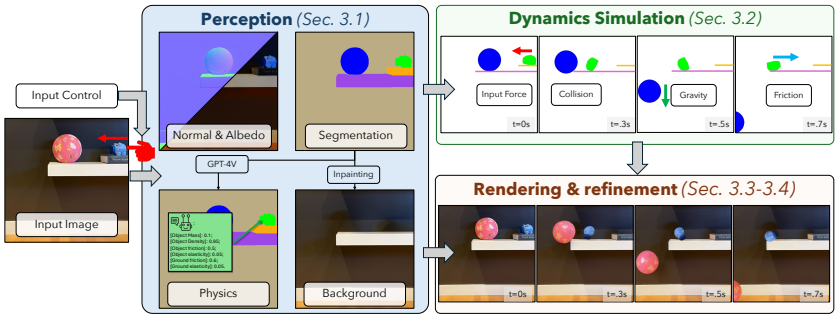

Fig. 2: Method overview. Our framework consists of three interleaved components: the perception module, the dynamics simulation module, and the rendering module. The perception module interprets the semantics, geometry, and physical parameters in the given image. The dynamics simulation module simulates the rigid-body motion and interactions of each instance in the scene, governed by Newton's Laws and physical constraints. The rendering module renders the final outcome, leveraging an off-the-shelf relighting model and diffusion-model-based video priors.

perspective, where our method takes a single image and user-specific initial force or torque as input, and models object dynamics explicitly via rigid physics simulation. Our key insight is that physical parameters can be inferred from a single image automatically through large foundation models such as GPT-4V [2, 57].

3 Approach

Given a single input image, our goal is to generate a realistic T-frame video featuring rigid body dynamics and interactions. Our generation can be conditioned on user-specific inputs in the form of an initial force $\mathbf{F}_0$ and/or torque $\boldsymbol{\tau}_0$. Our key insight is to incorporate physics-informed simulation into the generation process, ensuring the dynamics are realistic and interactive, thus producing large, plausible object motion. To achieve this, we present a novel framework consisting of three stages: physical understanding, dynamics generation, and generative rendering. The perception module aims to reason about the semantics, collision geometry, and physics of the objects presented in the image (Sec. 3.1). The dynamics module then leverages the input force/torque, as well as the inferred shape and geometry, to simulate the rigid-body motions of the objects, as well as object-object and object-scene interactions, such as friction and collisions (Sec. 3.2). Finally, we convert the simulated dynamics into pixel-level motion and combine image-based warping and generative video to render the final video (Sec. 3.3). Fig. 2 depicts the overall framework.

3.1 Perception

Simulating dynamics on an image requires a holistic understanding of object compositions, materials, geometry, and physics. Toward this goal, we designed

our perception module to infer such properties from a single image. Our key insight is to harness readily available, large pretrained models [2, 15, 27, 57, 62] to achieve this goal, as shown in Fig. 2.

Segmentation. Identifying and segmenting each individual physical entity lays the foundation for image-based dynamics. Inspired by the recent success of large pretrained segmentation models, we incorporate GPT-4V [2, 57] to recognize all image categories and send to Grounded-SAM [62] to detect and segment each individual instance-level objects $\{\mathbf{o}^i \in \mathbb{R}^{W\times H}\}_i^N$, where $\mathbf{o}^i$ is the binary mask for i-th object. We also query the GPT-4V to classify the objects into foreground and background based on their movability. The foreground objects are send to the physical simulation once user inputs are applied. We extract collision boundaries and supporting edges (e.g., ground, walls, etc.) from the background objects. The details are presented in the Appendix A.

Physical properties reasoning. Realistic physics dependents on accurate physical parameters, such as surface friction, mass, and elasticity. Unlike prior physics-based image dynamics work [20], our work leverages visual foundational models to reason physical properties directly. Our solution is simple yet effective. Inspired by the success of mask-based prompting [2, 57, 91], we directly ask GPT-4V [57] for certain physical properties, providing an object mask overlaid on the input image. Following [86, 95], we send the crafted prompt to GPT-4V, which contains GPT-4V instructions to return a quantitative measure of each queried property in a metric unit. For each object, we query its mass, elasticity, and friction coefficient (M_i, E_i, μ_i). We use the Coulomb friction model [60] for friction modeling. The elasticity coefficient is a scalar, where 0.0 results in no bounce, and 1.0 results in a perfect bounce. Given each object mass M_i in grams and its shape primitive, we compute the rotational inertia $\mathbf{I}_i$ accordingly.

Geometry primitives. Rasterized instance masks are not suitable for physical simulation; hence, we need to convert each object into vectorized shape primitives. We fit two types of primitive shapes: a circle and generic polygons. For each object, we choose the primitive type with the maximum coverage of its segmentation. For circles, we fit the center and radius to cover the segmentation mask. Circles allow us to realistically simulate rolling motions. For non-circle objects, we instead fit the generic polygons. Specifically, we perform contour extraction on the segmentation masks to obtain the polygon vertices.

Intrinsic decomposition. The dynamic movement of the objects will also result in comprehensive changes in shading, as their position wrt the lighting environment changes. To compensate for this effect during rendering, we must perform image decomposition to infer albedo, normal for each object $(\mathbf{A}_0^i, \mathbf{N}_0^i)$ and background scene lighting $\mathbf{L}$ from the single image. To achieve this we leverage off-the-shelf intrinsic decomposition model [15] to compute $\mathbf{A}_0^i$ and surface normal estimator [27] to compute $\mathbf{N}_0^i$. We model $\mathbf{L}$ as directional light source and estimate $\mathbf{L}$ using the optimization proposed in [16].

Background inpainting. Once the foreground objects move, they will leave holes in the background. To produce realistic and complete video footage, we leverage an off-the-shelf generative image inpainting model [94] to recover the complete background scene without foreground objects. The inpainted image $\mathbf{B} \in \mathbb{R}^{W \times H \times 3}$ will be used for composition in the rendering stage.

3.2 Image space dynamics simulation

Given the foreground objects with physical properties, we use rigid-body physics for dynamics simulation in image spaces. We choose to perform simulations in image spaces for three reasons: 1) image-space dynamics are better coupled with our output video; 2) a full 3D simulation requires complete 3D scene reconstruction and understanding, which remains an unsolved problem from a single image; 3) image-space dynamics can already cover various object dynamics and have been widely used in prior work [4, 20, 36, 47, 81]. Specifically, at a given time t, each rigid object i is characterized by its 2D pose and velocity at its center of mass. The position includes a translation $\mathbf{t}^i(t) \in \mathbb{R}^2$ and a rotation $\mathbf{R}^i(t) \in \mathbb{SO}(2)$ specified in a world coordinate. The velocity includes linear velocity $\boldsymbol{\nu}^i(t) \in \mathbb{R}^2$ and angular velocity $\boldsymbol{\omega}^i(t) \in \mathbb{R}^2$. Hence, the state of the each object i at time t can be represented as $\mathbf{q}^i(t) = \left[\mathbf{t}^i(t), \mathbf{R}^i(t), \boldsymbol{\nu}^i(t), \boldsymbol{\omega}^i(t)\right]$.

Following [26], the rigid body motion dynamics is given by Eq. (1) for each object. For simplicity, we omit the object index i in the following:

$$\frac{d}{dt}\mathbf{q}(t) = \frac{d}{dt}\begin{bmatrix}\mathbf{t}(t)\\ \mathbf{R}(t)\\ \boldsymbol{\nu}(t)\\ \boldsymbol{\omega}(t)\end{bmatrix} = \begin{bmatrix}\mathbf{v}(t)\\ \boldsymbol{\omega}(t) \times \mathbf{R}(t)\\ \frac{\mathbf{F}(t)}{M}\\ \mathbf{I}(t)^{-1}(\boldsymbol{\tau} - \boldsymbol{\omega}(t) \times \mathbf{I}(t)\boldsymbol{\omega}(t))\end{bmatrix} \tag{1}$$

where $\mathbf{F}$ is the force, $\boldsymbol{\tau}$ is the torque, M is the mass of the object, $\mathbf{I}(t) = \mathbf{R}(t)\mathbf{I}\mathbf{R}(t)^T$ is the rotation inertia in world coordinates, $\mathbf{I} \in \mathbb{R}^{2\times2}$ is the inertia matrix in its body coordinates. This inertia indicates its resistance against rotational motion. The state $\mathbf{q}(t)$ is given by integrating Eq. (1):

$$\mathbf{q}(t) = \mathbf{q}(0) + \int_0^t \frac{d}{dt}\mathbf{q}(t)dt = \mathbf{q}(0) + \sum_{i=1}^{T} \frac{d}{dt}\mathbf{q}(t)|_{t=t_i} \Delta_t \tag{2}$$

where $\mathbf{q}(0)$ is the initial condition specified in the input image. We compute the integral using numerical ODE integrations, *e.g.* Euler method [21]. Given the initial state of the objects, the physical motion simulation module synthesis each foreground object future motion. The location of each object is updated by the affine transformation $\mathbf{T}(t) = [\mathbf{R}(t), \mathbf{t}(t)]$ from the initial location.

External force and torque. For each object, the external forces $\mathbf{F}(t)$ and torques $\boldsymbol{\tau}$ include gravity, friction, and elasticity between the object's surface and the environment. Specifically, we consider the initial external force and torque from user input, as well as gravity, rolling friction, and sliding friction.

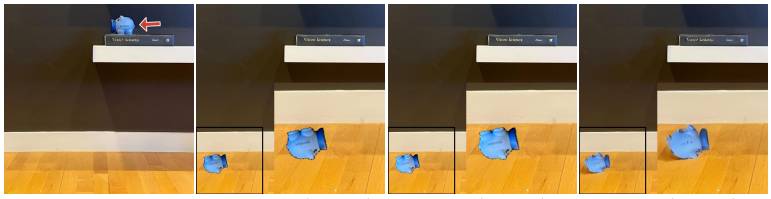

Fig. 3: Rendered video comparison. We show a toy example of piggy bank. The left shows the input frame, and the rest 3 are future frame generations. The composited piggy bank hasn't been aware of the light change. The relighted output synthesized no shadows beneath. The rendered output from diffusion model is most photo realistic.

Collision. When objects move, they might collide with each other or with the scene boundary (e.g., hitting a wall). Therefore, collision checking is necessary at every time step. A collision will result in offset forces being applied from its center of mass, producing $\boldsymbol{\tau}$ and causing the object to rotate. In collision reactions, changes in energy and momentum are minimized to adhere to Newton's conservation laws. In the case of an object falling, it could bounce back or start rolling on the ground due to the elasticity of both the object and the ground.

3.3 Rendering

Given the simulated motion sequence $\{\mathbf{q}^i(0), \ldots, \mathbf{q}^i(t), \ldots\}_i^N$ and the input image $\mathbf{C}$, the rendering module outputs the rendered video $\{\ldots, \mathbf{C}(t), \ldots\}$. There are a few desiderata for generating realistic video: it needs to be temporally consistent, complete, reflect the lighting changes as the object moves, and accurately represent the simulated image space dynamics. To achieve this goal, we design a novel motion-guided rendering pipeline consisting of an image-based transformation and composition module to ensure motion awareness, a relighting module to ensure the plausibility of lighting physics and a generative video editing module that retouches the composed and relit video to ensure realism.

Composited video. Given the object state from the physical motion module, we compose an initial video by alpha-blending the foreground scene with the static inpainted background from Sec. 3.1. The foreground scene is rendered by performing forward-warping from the input image to future frames using the affine transformation $\mathbf{T}(t)$. The alpha channel is computed using the same procedure, with the input being the segmentation mask in Sec. 3.1:

$$\hat{\mathbf{X}}(t) = \texttt{composite}(\mathbf{B}, \texttt{warp}(\mathbf{X}^0, \mathbf{T}^0(t)) \ldots \texttt{warp}(\mathbf{X}^i, \mathbf{T}^i(t)) \ldots ..)$$

where $\mathbf{X}^i$ are segmented input image of i-th object. Apart from the RGB sequence $\hat{\mathbf{V}} = \{\ldots \hat{\mathbf{X}}(t) \ldots\}$, we also use the same affine warping and composition to compute the blended albedo map $\hat{\mathbf{A}}(t)$ and surface normal $\hat{\mathbf{N}}(t)$ from $\hat{\mathbf{A}}(0)$ and $\hat{\mathbf{N}}(0)$ respectively, which is later used for relighting.

Relighting. Our relighting module simulates the changes in shading due to object movement. Specifically, it takes the RGB image $\hat{\mathbf{X}}(t)$, the transformed albedo $\hat{\mathbf{A}}(t)$ and surface normal $\hat{\mathbf{N}}(t)$ at each time t, as well as the estimated directional light $\mathbf{L}$ as input. We then perform reshading using the the Lambertian shading model f for foreground objects: $\tilde{\mathbf{X}}(t) = f(\hat{\mathbf{X}}(t), \hat{\mathbf{A}}(t), \hat{\mathbf{N}}(t), \mathbf{L})$, which returns the relit image $\tilde{\mathbf{X}}(t)$ for each frame t.

3.4 Generative refinement with latent diffusion models

One limitation of the single-image based relighting model is that it neglects the background and cannot handle complicated lighting effects due to complex object-object interactions, such as ambient occlusion and cast shadows. Moreover, the composition results in unnatural boundaries. To address these issues and enhance realism, we incorporated a diffusion-based video to refine the relit video, denoted as $\tilde{\mathbf{V}} = \{\dots, \tilde{\mathbf{X}}(t), \dots\}$, and obtain the final video output.

Specifically, we incorporate a pretrained video diffusion model to refine our video $\tilde{\mathbf{V}}$. We first use the pretrained latent diffusion encoder [63] to encode the guidance video $\tilde{\mathbf{V}}$ into a latent code $\mathbf{z}$. Inspired by SDEdit [54], we add noise to the guided latent code and gradually denoise it using the pretrained video diffusion. Given that the content in the guided latent code is already satisfactory, we do not perform the denoising process from scratch. Instead, we define an initial noise strength s where $s \in [0, 1]$ controls the amount of noise added to $\mathbf{z}$. In practice, we find that a certain approach finds a good trade-off between fidelity and realism. At each denoising step, we ensure the foreground objects are as consistent as possible with the guidance (copied from the guided latent code) while synthesizing new content (e.g., shadows) in the background (using the denoised latent code). To this end, we use a fusion weight w to control how much of the generated latent to inject into the foreground; the fusion weight is gradually increased during the denoising as the perturbation of noise decreases. We also define a fusion timestamp δ as the signal to stop the fusion. The final denoised latent code $\mathbf{z}_*$ is then decoded to our final video output $\mathbf{V}$. Please see Appendix A.3 for a detailed algorithm.

Fig. 3 depicts the composed video $\hat{\mathbf{V}}$, the relit video $\tilde{\mathbf{V}}$, and our final output $\mathbf{V}$, illustrating the effectiveness of each rendering component.

4 Experiments

Implementation details. For physical body simulation, we use a 2D rigid body physics simulator Pymunk [13]. For each experiment, we run 120 simulation steps and uniformly sample 16 frames to form the videos. We set the generation resolution at 512×512 so that frames can easily fit into the diffusion models. For video diffusion prior model, we use SEINE [19]. For inference, DDIM sampling [68] is used and the noise strength i set to $s = 0.5$, *i.e.*, executing $25/50$ steps for denoising. The latents fusion stops at timestamp $\delta = 5$.

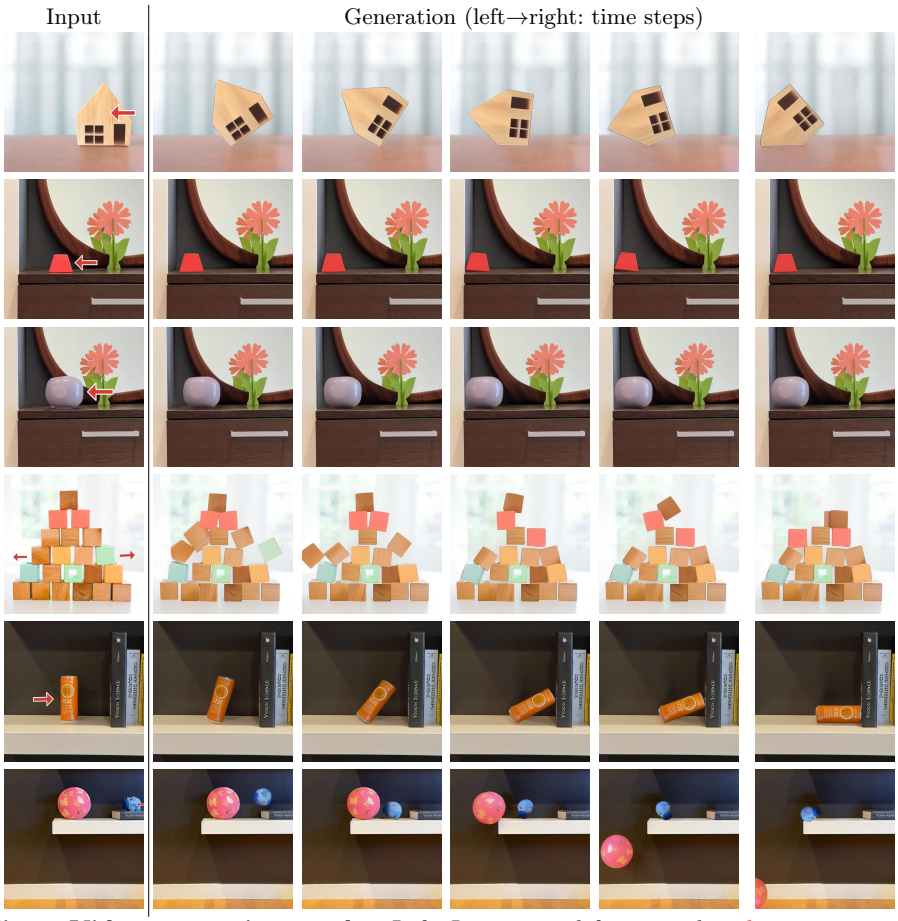

Fig. 4: Video generation results. Left: Input initial frame with red arrows representing the initial force or speed on the object. Right: Generated future frames.

Speed. PhysGen's run-time speed is 3 minutes for one image-to-video gen on a Nvidia A40 GPU. Image understanding and relighting takes the most time.

Data. To show the generalizability and robustness of our method, we use both internet data and self-captured indoor images from a cell phone. The dataset is diverse, with variations in lighting, object count, geometries, physical attributes, and environmental boundaries. We compare different approaches on 15 photos that require physical reasoning. Additionally, we collect 50 recorded videos of a given scene as ground truth by varying input conditions for quantitative comparison. Randomly selected sequences are shown in Appendix B.2.

Baselines. We compare against two streams of approaches. The first stream is current state-of-the-art I2V models: SEINE [19], I2VGen-XL [96] and Dynam-

Methods	**Physical-realism↑**	**Photo-realism↑**
SEINE [19]	1.39	1.86
DynamiCrafter [88]	1.68	1.81
I2VGen-XL [96]	2.11	2.25
Ours	**4.14**	**3.86**

Table 1: Human evaluation. We evaluate both physical-realism and photo-realism of the three competing image-to-video models. The user is asked to evaluate the generated video using a 5-point scale from strongly disagree (1) to strongly agree(5) that the video is physical-realistic and photo-realistic. Our method ranks the first in both terms.

Methods	**Image-FID↓**	**Motion-FID↓**
SEINE [19]	138.89	38.76
DynamiCrafter [88]	**99.64**	109.61
I2VGen-XL [96]	104.62	82.04
Ours	105.70	**30.20**

Table 2: Quantitative evaluation. We measure the Image-FID and Motion-FID of videos generated by four methods against the collected GT videos for the given scene. Our method achieves both low Image-FID and Motion-FID.

iCrafter [88]. These methods take both an image and a text prompt as input for generation. The text prompt is generated by GPT-4V [57] to describe the future events. Another stream we compare is the image-based manipulation approach Motion Guidance [29]. This method uses an image and optical flow as inputs to predict future movement and generate corresponding images. We employ computed motion flow from our physical motion module as the target input for Motion Guidance to optimize each video frame.

4.1 Results

We show our generated results on 8 different physical procedures from multiple sources in Fig. 4. Our method could simulate reasonable complex physical procedures which involves object-object interaction, object-scene interaction with different physical properties and initial conditions.

Visual comparisons. The comparison results are shown in Fig. 5. As can be seen, image-video generation methods could not produce any physically plausible futures. For each method, even we carefully tune the input prompt, the models couldn't understand the direction, physical meanings and interactions. All 3 models fail to synthesize realistic videos. For motion guidance with accurate motion flow provided, the results is better but brings a lot inconsistency. When there are multiple objects interaction, the method suffer from introducing new contents. For the domino case, the generation is not smooth and contain artifacts as it could not accurately model the physical procedure.

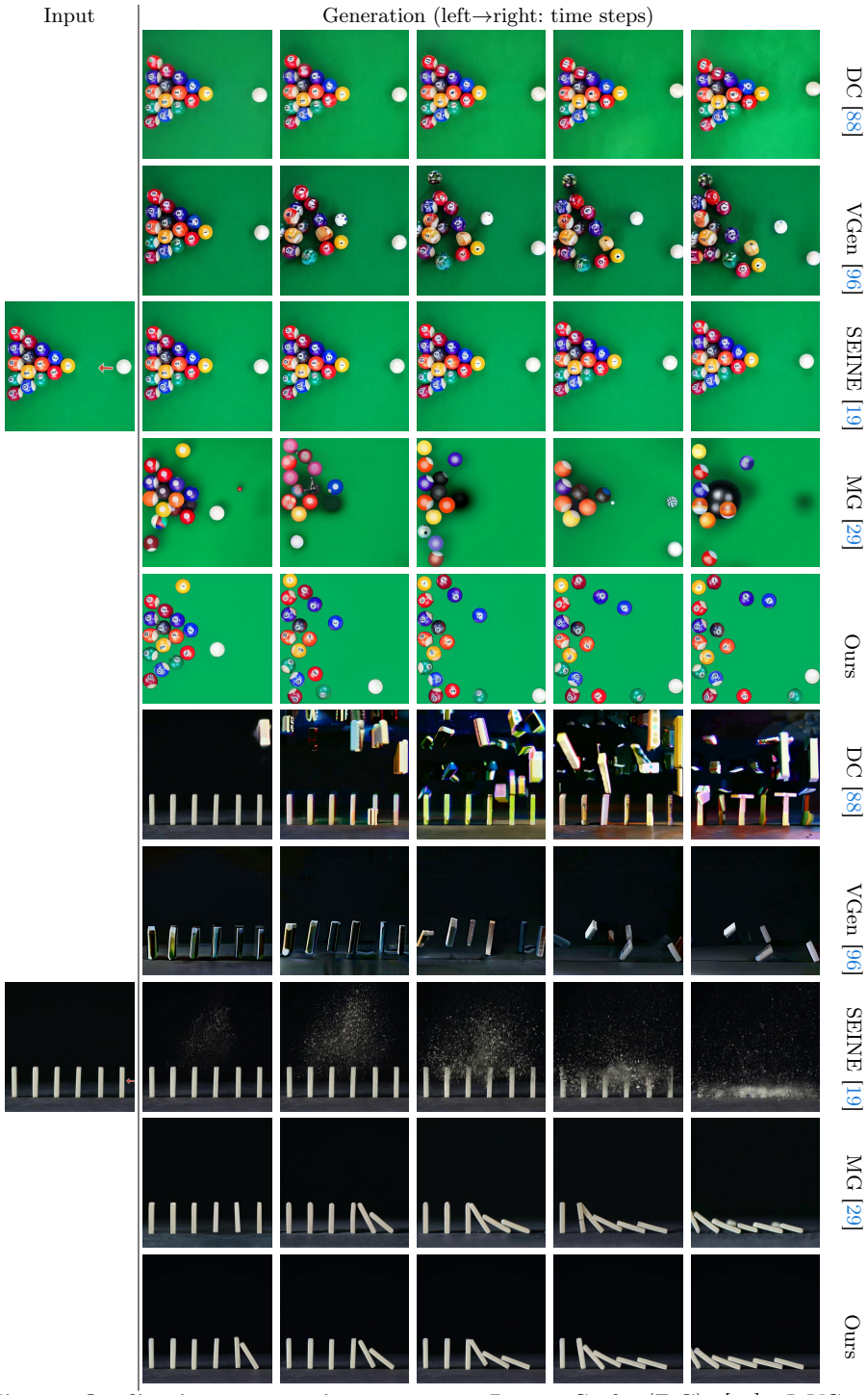

Fig. 5: Qualitative comparison. against DynamiCrafter(**DC**) [88], I2VGen-XL(**VGen**) [96], **SEINE** [19], Motion Guidance(**MG**) [29].

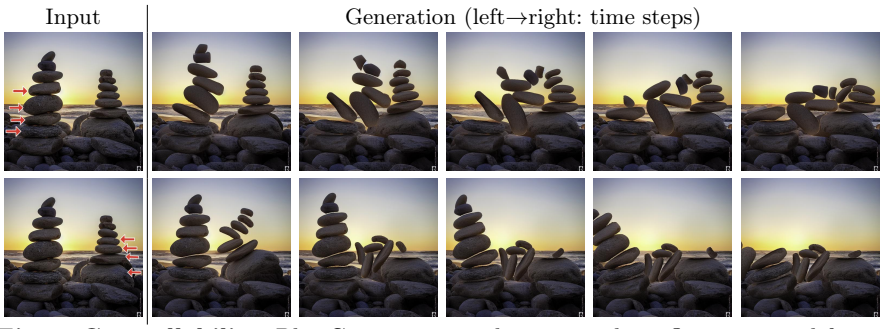

Fig. 6: Controllability. PhysGen generates diverse results reflecting initial forces.

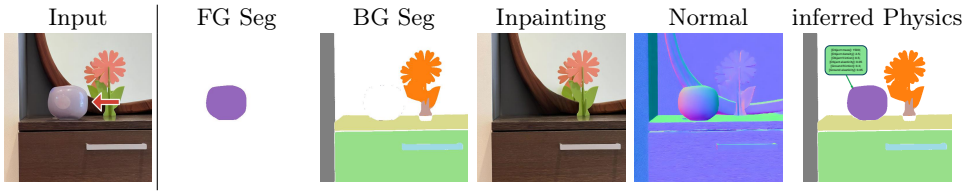

Fig. 7: Qualitative results from the perception module. The results include foreground and background mask, inpainting output, normal map and inferred physics.

4.2 Human evaluation

To comprehensively assess both the **physical-realism** and **photo-realism** of the generated videos, we conducted human evaluations following a similar methodology to prior work [17,79,84,92]. Here by physical-realism we mean whether the content of the generated video is realistic in its dynamics and object interactions and accurately reflects the instructed movements.

Specifically, we include 15 videos with diverse objects and conditions for evaluation, compared against other three I2V model generations (SEINE [19], DynamiCrafter [88], I2VGen-XL [96]). For each compared baseline method, we meticulously adjusted the input prompts and conducted multiple runs to select the most plausible output. In contrast, our method did not undergo specific tuning for each video. We randomized the presentation order and asked 14 participants to rate the videos on a five-point scale, from strongly disagree (1) to strongly agree (5) for physical-realism and photo-realism. The results in Tab. 1 show that our method achieves the highest ratings. Our average scores for physical-realism (4.14) and photo-realism (3.86) fall within the Agree level, while other methods perform poorly.

4.3 Quantitative evaluation

We quantitatively evaluate the collected GT videos of a given scene. The GT data includes 833 images. For each method, we generate 50 videos with random

samples, collecting 800 images per method. Our method varies initial conditions to ensure all generated videos are different. Following [12], we adopt the Fréchet Inception Distance (FID) [33] as evaluation metric. We didn't use Fréchet Video Distance (FVD) [74] due to the small sample size and the goal to disentangle evaluation of appearance and motion. We evaluate appearance using **Image-FID** and motion using **Motion-FID**. For Motion-FID, we use RAFT [72] to extract and colorize optical flow [5], then compute FID from these rendered images. The results are shown in Tab. 2. We notice other methods which either generate minimal motion, leading to low Image-FID and high Motion-FID (e.g., DynamiCrafter [88]), or produce realistic motion but fail to maintain image consistency, resulting in low Motion-FID and high Image-FID (e.g., SEINE [19]). Our approach achieves both low Image-FID and Motion-FID compared to others.

4.4 Analysis

Perception evaluation. We evaluate the perception module using 10 complex open-world images with 118 annotated movable instances (COCO [48] format). Our system achieves **0.93** precision, **0.82** recall at 0.5 IoU. More details are shown in the Appendix C. Fig. 7 shows intermediate outputs from our perception module and simulation module.

Physical reasoning. We compare the physical parameters estimated by GPT-4V to random values sampled from human-estimated ranges. Without GPT-4V, Image-FID increases from **105.70** to **111.01**, Motion-FID from **30.20** to **36.60**. It shows the physical reasoning are crucial for appearance and motion realism.

Controllability. We showcase the controllability and diversity of our method in Fig. 6 by varying the initial conditions. For different rows, we adjust the direction and magnitude of initial speed and forces. The results show diverse physical-plausible trajectories. This demonstrates the potential of applying our method for diverse physical world generations.

Limitations. Our method is an initial step towards physics-based video generation but has two main limitations: it focuses mainly on rigid objects, limiting its application to non-rigid ones, and it lacks a comprehensive 3D understanding, making it unable to handle out-of-plane motions. We leave leveraging deformable physics and full 3D understanding as future work.

5 Conclusion

We present PhysGen, a novel image-to-video generation method that turns a static image into a high-fidelity, physics-grounded, and temporally coherent videos. The main module is a model-based physical simulator with a data-driven video generation process. PhysGen enables a series of controllable and interactive downstream applications. PhysGen provide a new image-to-video generation paradigm and hopefully can inspire more follow-up works.

Acknowledgements

This project is supported by NSF Awards #2331878, #2340254, and #2312102, the IBM IIDAI Grant, and an Intel Research Gift. We greatly appreciate the NCSA for providing computing resources. We thank Tianhang cheng for helpful discussions. We thank Emily Chen and Gloria Wang for proofreading.

References

1. Aberman, K., Weng, Y., Lischinski, D., Cohen-Or, D., Chen, B.: Unpaired motion style transfer from video to animation. ACM TOG (2020)
2. Achiam, J., Adler, S., Agarwal, S., Ahmad, L., Akkaya, I., Aleman, F.L., Almeida, D., Altenschmidt, J., Altman, S., Anadkat, S., et al.: Gpt-4 technical report. arXiv preprint arXiv:2303.08774 (2023)
3. Ajay, A., Bauza, M., Wu, J., Fazeli, N., Tenenbaum, J.B., Rodriguez, A., Kaelbling, L.P.: Combining Physical Simulators and Object-Based Networks for Control. In: ICRA (2019)
4. Ajay, A., Wu, J., Fazeli, N., Bauza, M., Kaelbling, L.P., Tenenbaum, J.B., Rodriguez, A.: Augmenting Physical Simulators with Stochastic Neural Networks: Case Study of Planar Pushing and Bouncing. In: IROS (2018)
5. Baker, S., Scharstein, D., Lewis, J.P., Roth, S., Black, M.J., Szeliski, R.: A database and evaluation methodology for optical flow. IJCV (2011)
6. Bar-Tal, O., Chefer, H., Tov, O., Herrmann, C., Paiss, R., Zada, S., Ephrat, A., Hur, J., Li, Y., Michaeli, T., et al.: Lumiere: A space-time diffusion model for video generation. arXiv preprint arXiv:2401.12945 (2024)
7. Battaglia, P., Pascanu, R., Lai, M., Jimenez Rezende, D., et al.: Interaction networks for learning about objects, relations and physics. NeurIPS (2016)
8. 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)
9. Blattmann, A., Milbich, T., Dorkenwald, M., Ommer, B.: ipoke: Poking a still image for controlled stochastic video synthesis. In: ICCV (2021)
10. Blattmann, A., Milbich, T., Dorkenwald, M., Ommer, B.: Understanding object dynamics for interactive image-to-video synthesis. In: CVPR (2021)
11. Blattmann, A., Milbich, T., Dorkenwald, M., Ommer, B.: Understanding object dynamics for interactive image-to-video synthesis. In: CVPR (2021)
12. Blattmann, A., Rombach, R., Ling, H., Dockhorn, T., Kim, S.W., Fidler, S., Kreis, K.: Align your latents: High-resolution video synthesis with latent diffusion models. In: CVPR (2023)
13. Blomqvist, V.: Pymunk. https://pymunk.org (2023)
14. Bowen, R.S., Tucker, R., Zabih, R., Snavely, N.: Dimensions of motion: Monocular prediction through flow subspaces. In: 3DV (2022)
15. Careaga, C., Aksoy, Y.: Intrinsic image decomposition via ordinal shading. TOG (2023)
16. Careaga, C., Miangoleh, S.M.H., Aksoy, Y.: Intrinsic harmonization for illumination-aware compositing. In: SIGGRAPH Asia (2023)
17. 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)

18. Chen, X., Liu, Z., Chen, M., Feng, Y., Liu, Y., Shen, Y., Zhao, H.: Livephoto: Real image animation with text-guided motion control. arXiv preprint arXiv:2312.02928 (2023)
19. Chen, X., Wang, Y., Zhang, L., Zhuang, S., Ma, X., Yu, J., Wang, Y., Lin, D., Qiao, Y., Liu, Z.: Seine: Short-to-long video diffusion model for generative transition and prediction. arXiv preprint arXiv:2310.20700 (2023)
20. Chuang, Y.Y., Goldman, D.B., Zheng, K.C., Curless, B., Salesin, D.H., Szeliski, R.: Animating pictures with stochastic motion textures. ACM TOG (2005)
21. Ciarlet, P.G., Lions, J.L.: Handbook of numerical analysis. Gulf Professional Publishing (1990)
22. Davis, A., Bouman, K.L., Chen, J.G., Rubinstein, M., Durand, F., Freeman, W.T.: Visual vibrometry: Estimating material properties from small motion in video. In: CVPR (2015)
23. Davis, A., Chen, J.G., Durand, F.: Image-space modal bases for plausible manipulation of objects in video. ACM TOG (2015)
24. Eftekhar, A., Sax, A., Malik, J., Zamir, A.: Omnidata: A scalable pipeline for making multi-task mid-level vision datasets from 3d scans. In: ICCV (2021)
25. Endo, Y., Kanamori, Y., Kuriyama, S.: Animating landscape: self-supervised learning of decoupled motion and appearance for single-image video synthesis. arXiv preprint arXiv:1910.07192 (2019)
26. Friedland, B.: Control system design: an introduction to state-space methods. Courier Corporation (2012)
27. Fu, X., Yin, W., Hu, M., Wang, K., Ma, Y., Tan, P., Shen, S., Lin, D., Long, X.: Geowizard: Unleashing the diffusion priors for 3d geometry estimation from a single image. arXiv preprint arXiv:2403.12013 (2024)
28. Ge, S., Nah, S., Liu, G., Poon, T., Tao, A., Catanzaro, B., Jacobs, D., Huang, J.B., Liu, M.Y., Balaji, Y.: Preserve your own correlation: A noise prior for video diffusion models. In: ICCV (2023)
29. Geng, D., Owens, A.: Motion guidance: Diffusion-based image editing with differentiable motion estimators. In: ICLR (2024)
30. Girdhar, R., Singh, M., Brown, A., Duval, Q., Azadi, S., Rambhatla, S.S., Shah, A., Yin, X., Parikh, D., Misra, I.: Emu video: Factorizing text-to-video generation by explicit image conditioning. arXiv preprint arXiv:2311.10709 (2023)
31. Guo, Y., Yang, C., Rao, A., Liang, Z., Wang, Y., Qiao, Y., Agrawala, M., Lin, D., Dai, B.: Animatediff: Animate your personalized text-to-image diffusion models without specific tuning (2023)
32. Gupta, A., Yu, L., Sohn, K., Gu, X., Hahn, M., Fei-Fei, L., Essa, I., Jiang, L., Lezama, J.: Photorealistic video generation with diffusion models. arXiv preprint arXiv:2312.06662 (2023)
33. Heusel, M., Ramsauer, H., Unterthiner, T., Nessler, B., Hochreiter, S.: Gans trained by a two time-scale update rule converge to a local nash equilibrium. NeurIPS (2017)
34. 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)
35. Ho, J., Jain, A., Abbeel, P.: Denoising diffusion probabilistic models. NeurIPS (2020)
36. Holynski, A., Curless, B.L., Seitz, S.M., Szeliski, R.: Animating pictures with eulerian motion fields. In: CVPR (2021)
37. Hu, Y., Anderson, L., Li, T.M., Sun, Q., Carr, N., Ragan-Kelley, J., Durand, F.: Difftaichi: Differentiable programming for physical simulation. ICLR (2020)

38. 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 TOG (2018)
39. 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 TOG (2019)
40. Jhou, W.C., Cheng, W.H.: Animating still landscape photographs through cloud motion creation. IEEE Transactions on Multimedia (2015)
41. Kirillov, A., Mintun, E., Ravi, N., Mao, H., Rolland, C., Gustafson, L., Xiao, T., Whitehead, S., Berg, A.C., Lo, W.Y., Dollár, P., Girshick, R.: Segment anything. arXiv:2304.02643 (2023)
42. Koenig, N., Howard, A.: Design and use paradigms for gazebo, an open-source multi-robot simulator. In: IROS (2004)
43. Kondratyuk, D., Yu, L., Gu, X., Lezama, J., Huang, J., Hornung, R., Adam, H., Akbari, H., Alon, Y., Birodkar, V., et al.: Videopoet: A large language model for zero-shot video generation. arXiv preprint arXiv:2312.14125 (2023)
44. Li, X., Qiao, Y.L., Chen, P.Y., Jatavallabhula, K.M., Lin, M., Jiang, C., Gan, C.: Pac-nerf: Physics augmented continuum neural radiance fields for geometry-agnostic system identification. In: ICLR (2023)
45. Li, Y., Lin, T., Yi, K., Bear, D., Yamins, D.L., Wu, J., Tenenbaum, J.B., Torralba, A.: Visual grounding of learned physical models. In: ICML (2020)
46. Li, Y., Wu, J., Tedrake, R., Tenenbaum, J.B., Torralba, A.: Learning particle dynamics for manipulating rigid bodies, deformable objects, and fluids. In: ICLR (2019)
47. Li, Z., Tucker, R., Snavely, N., Holynski, A.: Generative image dynamics. arXiv preprint arXiv:2309.07906 (2023)
48. Lin, T.Y., Maire, M., Belongie, S., Hays, J., Perona, P., Ramanan, D., Dollár, P., Zitnick, C.L.: Microsoft coco: Common objects in context. In: ECCV (2014)
49. Liu, S., Zeng, Z., Ren, T., Li, F., Zhang, H., Yang, J., Li, C., Yang, J., Su, H., Zhu, J., et al.: Grounding dino: Marrying dino with grounded pre-training for open-set object detection. arXiv preprint arXiv:2303.05499 (2023)
50. Liu, T., Bargteil, A.W., O'Brien, J.F., Kavan, L.: Fast simulation of mass-spring systems. ACM TOG (2013)
51. Lv, J., Huang, Y., Yan, M., Huang, J., Liu, J., Liu, Y., Wen, Y., Chen, X., Chen, S.: Gpt4motion: Scripting physical motions in text-to-video generation via blender-oriented gpt planning. In: CVPRW. pp. 1430–1440 (2024)
52. Mahapatra, A., Kulkarni, K.: Controllable animation of fluid elements in still images. In: CVPR (2022)
53. Mallya, A., Wang, T.C., Liu, M.Y.: Implicit warping for animation with image sets. In: NeurIPS (2022)
54. Meng, C., He, Y., Song, Y., Song, J., Wu, J., Zhu, J.Y., Ermon, S.: SDEdit: Guided image synthesis and editing with stochastic differential equations. In: International Conference on Learning Representations (2022)
55. Mrowca, D., Zhuang, C., Wang, E., Haber, N., Fei-Fei, L.F., Tenenbaum, J., Yamins, D.L.: Flexible neural representation for physics prediction. In: NeurIPS (2018)
56. NVIDIA: Nvidia physx. https://developer.nvidia.com/physx-sdk (2019)
57. OpenAI: Gpt-4v(ision) system card (2023)
58. OpenAI: Creating video from text. https://openai.com/sora (2024)
59. Peebles, W., Xie, S.: Scalable diffusion models with transformers. arXiv preprint arXiv:2212.09748 (2022)

60. Popova, E., Popov, V.L.: The research works of coulomb and amontons and generalized laws of friction. Friction (2015)
61. Qiu, H., Chen, Z., Jiang, Y., Zhou, H., Fan, X., Yang, L., Wu, W., Liu, Z.: Relitalk: Relightable talking portrait generation from a single video (2023)
62. Ren, T., Liu, S., Zeng, A., Lin, J., Li, K., Cao, H., Chen, J., Huang, X., Chen, Y., Yan, F., Zeng, Z., Zhang, H., Li, F., Yang, J., Li, H., Jiang, Q., Zhang, L.: Grounded sam: Assembling open-world models for diverse visual tasks (2024)
63. Rombach, R., Blattmann, A., Lorenz, D., Esser, P., Ommer, B.: High-resolution image synthesis with latent diffusion models. In: CVPR (2022)
64. Sanchez-Gonzalez, A., Heess, N., Springenberg, J.T., Merel, J., Riedmiller, M., Hadsell, R., Battaglia, P.: Graph networks as learnable physics engines for inference and control. In: ICML (2018)
65. Schödl, A., Szeliski, R., Salesin, D.H., Essa, I.: Video textures. In: PACMCGIT (2000)
66. Singer, U., Polyak, A., Hayes, T., Yin, X., An, J., Zhang, S., Hu, Q., Yang, H., Ashual, O., Gafni, O., et al.: Make-a-video: Text-to-video generation without text-video data. arXiv preprint arXiv:2209.14792 (2022)
67. Sohl-Dickstein, J., Weiss, E., Maheswaranathan, N., Ganguli, S.: Deep unsupervised learning using nonequilibrium thermodynamics. In: ICML (2015)
68. Song, J., Meng, C., Ermon, S.: Denoising diffusion implicit models. arXiv preprint arXiv:2010.02502 (2020)
69. Song, Y., Ermon, S.: Generative modeling by estimating gradients of the data distribution. NeurIPS (2019)
70. Sugimoto, R., He, M., Liao, J., Sander, P.V.: Water simulation and rendering from a still photograph. In: SIGGRAPH Asia (2022)
71. Szummer, M., Picard, R.W.: Temporal texture modeling. In: ICIP (1996)
72. Teed, Z., Deng, J.: Raft: Recurrent all-pairs field transforms for optical flow. In: ECCV. pp. 402–419 (2020)
73. Todorov, E., Erez, T., Tassa, Y.: Mujoco: A physics engine for model-based control. In: IROS (2012)
74. Unterthiner, T., Van Steenkiste, S., Kurach, K., Marinier, R., Michalski, M., Gelly, S.: Towards accurate generative models of video: A new metric & challenges. arXiv preprint arXiv:1812.01717 (2018)
75. Villegas, R., Babaeizadeh, M., Kindermans, P.J., Moraldo, H., Zhang, H., Saffar, M.T., Castro, S., Kunze, J., Erhan, D.: Phenaki: Variable length video generation from open domain textual description. arXiv preprint arXiv:2210.02399 (2022)
76. Wang, X., Yuan, H., Zhang, S., Chen, D., Wang, J., Zhang, Y., Shen, Y., Zhao, D., Zhou, J.: Videocomposer: Compositional video synthesis with motion controllability. arXiv preprint arXiv:2306.02018 (2023)
77. Wang, Y., Chen, X., Ma, X., Zhou, S., Huang, Z., Wang, Y., Yang, C., He, Y., Yu, J., Yang, P., et al.: Lavie: High-quality video generation with cascaded latent diffusion models. arXiv preprint arXiv:2309.15103 (2023)
78. Wei, L.Y., Levoy, M.: Fast texture synthesis using tree-structured vector quantization. In: PACMCGIT (2000)
79. Wei, Y., Zhang, S., Qing, Z., Yuan, H., Liu, Z., Liu, Y., Zhang, Y., Zhou, J., Shan, H.: Dreamvideo: Composing your dream videos with customized subject and motion. arXiv preprint arXiv:2312.04433 (2023)
80. Weng, C.Y., Curless, B., Kemelmacher-Shlizerman, I.: Photo wake-up: 3d character animation from a single photo. In: CVPR (2019)
81. Wu, J., Lim, J.J., Zhang, H., Tenenbaum, J.B., Freeman, W.T.: Physics 101: Learning physical object properties from unlabeled videos. In: BMVC (2016)

82. Wu, J., Lu, E., Kohli, P., Freeman, W.T., Tenenbaum, J.B.: Learning to see physics via visual de-animation. In: NeurIPS (2017)
83. Wu, J., Yildirim, I., Lim, J.J., Freeman, W.T., Tenenbaum, J.B.: Galileo: Perceiving physical object properties by integrating a physics engine with deep learning. In: NeurIPS (2015)
84. Wu, R., Chen, L., Yang, T., Guo, C., Li, C., Zhang, X.: Lamp: Learn a motion pattern for few-shot-based video generation. arXiv preprint arXiv:2310.10769 (2023)
85. Xia, H., Lin, Z.H., Ma, W.C., Wang, S.: Video2game: Real-time, interactive, realistic and browser-compatible environment from a single video. In: CVPR (2024)
86. Xia, H., Lin, Z.H., Ma, W.C., Wang, S.: Video2game: Real-time, interactive, realistic and browser-compatible environment from a single video. In: CVPR (2024)
87. Xie, T., Zong, Z., Qiu, Y., Li, X., Feng, Y., Yang, Y., Jiang, C.: Physgaussian: Physics-integrated 3d gaussians for generative dynamics. In: CVPR (2024)
88. Xing, J., Xia, M., Zhang, Y., Chen, H., Yu, W., Liu, H., Wang, X., Wong, T.T., Shan, Y.: Dynamicrafter: Animating open-domain images with video diffusion priors. arXiv preprint arXiv:2310.12190 (2023)
89. Xu, Z., Wu, J., Zeng, A., Tenenbaum, J.B., Song, S.: Densephysnet: Learning dense physical object representations via multi-step dynamic interactions. In: RSS (2019)
90. Xue, T., Wu, J., Bouman, K.L., Freeman, W.T.: Visual dynamics: Stochastic future generation via layered cross convolutional networks. T-PAMI (2018)
91. Yang, J., Zhang, H., Li, F., Zou, X., Li, C., Gao, J.: Set-of-mark prompting unleashes extraordinary visual grounding in gpt-4v. arXiv preprint arXiv:2310.11441 (2023)
92. Yu, J., Cun, X., Qi, C., Zhang, Y., Wang, X., Shan, Y., Zhang, J.: Animatezero: Video diffusion models are zero-shot image animators. arXiv preprint arXiv:2312.03793 (2023)
93. Yu, L., Lezama, J., Gundavarapu, N.B., Versari, L., Sohn, K., Minnen, D., Cheng, Y., Gupta, A., Gu, X., Hauptmann, A.G., et al.: Language model beats diffusion–tokenizer is key to visual generation. arXiv preprint arXiv:2310.05737 (2023)
94. Yu, T., Feng, R., Feng, R., Liu, J., Jin, X., Zeng, W., Chen, Z.: Inpaint anything: Segment anything meets image inpainting. arXiv preprint arXiv:2304.06790 (2023)
95. Zhai, A.J., Shen, Y., Chen, E., Wang, G., Wang, X., Wang, S., Wang, X., Guan, K., Wang, S.: Physical property understanding from language-embedded feature fields. In: CVPR (2024)
96. Zhang, S., Wang, J., Zhang, Y., Zhao, K., Yuan, H., Qing, Z., Wang, X., Zhao, D., Zhou, J.: I2vgen-xl: High-quality image-to-video synthesis via cascaded diffusion models. arXiv preprint arXiv:2311.04145 (2023)
97. Zhao, J., Zhang, H.: Thin-plate spline motion model for image animation. In: CVPR (2022)

Supplementary Material – PhysGen: Rigid-Body Physics-Grounded Image-to-Video Generation

Shaowei Liu, Zhongzheng Ren, Saurabh Gupta*, and Shenlong Wang*

University of Illinois Urbana-Champaign
https://stevenlsw.github.io/physgen/

Abstract. The supplementary material provides implementations, additional analysis and experiments, as well as discussion of limitations in detail. In summary, we include

- Appendix A. More implementation details of each module in our pipeline.
- Appendix B. More experiment details besides the main paper.
- Appendix C. More experiments on key components in our framework.
- Appendix D. Additional qualitative comparisons and controllable generation results.
- Appendix E. Generation limitation analysis of our current approach.

A Implementation Details

A.1 Perception

Segmentation. We use GPT-4V to recognize all objects in the given image and determine if they are movable. Movable objects are treated as foreground objects, while non-movable objects are treated as background objects. The query prompt is shown in Fig. 1. The outputs from GPT-4V are sent to Grounded-SAM [62] for instance segmentation. We also enable non-maximum suppression (NMS) to prevent overlapping segmentation. For foreground objects, we check if the segmentation is fully connected within its mask. If not, we rerun Grounded-SAM to further separate any poor segmentation from the first round.

Boundary extraction. To extract boundaries from non-movable background objects, we utilize depth and normal information estimated from GeoWizard [27]. We first order the background objects according to relative depth estimation and select only those whose depth range falls within that of the foreground objects. For candidate objects, we extract their segmentation boundaries within the foreground object's depth range and use corresponding normal to determine if they are planes. If so, we fit horizontal or vertical edges to the boundaries and use it as the physical boundary of the scene.

* Equal advising

User: Describe all unique object categories in the given image, ensuring all pixels are included and assigned to one of the categories, do not miss any movable or static object appeared in the image, each category name is a single word and in singular noun format, do not include '-' in the name. Different categories should not be repeated or overlapped with each other in the image. For each category, judge if the instances in the image is movable, the answer is True or False. If there are multiple instances of the same category in the image, the judgement is True only if the object category satisfies the following requirements: 1. The object category is things (objects with a well-defined shape, e.g. car, person) and not stuff (amorphous background regions, e.g. grass, sky, largest segmentation component). 2. All instances in the image of this category are movable with complete shape and fully-visible.
Format Requirement: You must provide your answer in the following JSON format, as it will be parsed by a code script later. Your answer must look like: "category-1": False, "category-2": True
Do not include any other text in your answer. Do not include unnecessary words besides the category name and True/False values.

Fig. 1: Prompt used for GPT-4V image recognition and movability judgment.

Physical properties reasoning. The query prompt for reasoning the physical properties of a foreground object is shown in Fig. 2.

Geometry primitives. Given a object segmentation mask, we automatically choose the proper primitive that best fits the object. We first use a circle to fit the corresponding segmentation mask and compute the Intersection over Union (IoU) between the fitted mask and segmentation mask. If IoU is smaller than 0.85, we switch to the generic polygons by extracting the contour of the segmentation.

Background Inpainting. Given the foreground masks of the input image, we use off-the-shelf image inpainting model [94] to recover the background scene. Considering foreground objects might have shadow beneath, we dilate the foreground segmentation mask by a kernel size of 40 pixels. To this end, we aim to get a clean background image without shadows. However, if the input image is heavy-shadowed, the inpainting model could not remove it completely. We discuss it use a detailed example in Appendix E.

A.2 Image Space Dynamics Simulation

In image space dynamics simulation, we set Δ_t as 1 second, gravity as $g = 980cm/s^2$. Considering most foreground objects are captured at a similar distance, we set 1pixel as 1cm to map from image space to world space without reasoning metric scale. We visualize the physical simulation procedure of billiard balls and blocks with two different primitives (circle and polygon) in Fig. 3.

User: You will be given an image and a binary mask specifying an object on the image, analyze and provide your final answer of the object physical property. The query object will be enclosed in white mask. The physical property includes the mass, the friction and elasticity. The mass is in grams. The friction uses the Coulomb friction model, a value of 0.0 is frictionless. The elasticity value of 0.0 gives no bounce, while a value of 1.0 will give a perfect bounce.
Format Requirement: You must provide your answer in the following JSON format, as it will be parsed by a code script later. Your answer must look like: "mass": number, "friction": number, "elasticity": number The answer should be one exact number for each property, do not include any other text in your answer, as it will be parsed by a code script later.

Fig. 2: Prompt used for GPT-4V image recognition and movability judgment.

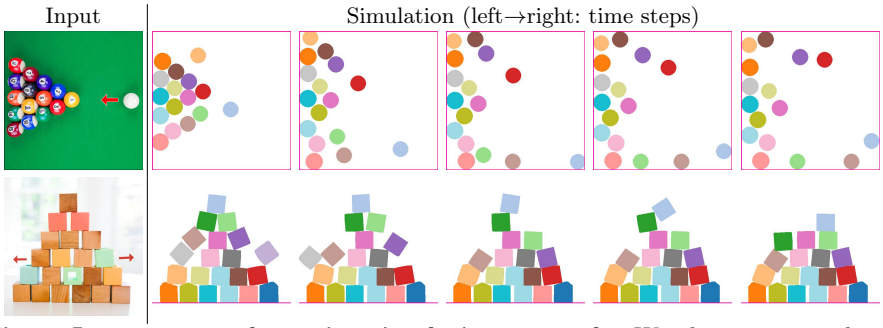

Fig. 3: Image space dynamics simulation example. We show two simulation examples with different primitives. The primitives are fitted from the segmentation mask of the input image and we perform dynamic simulation on those primitives. The edges show the physical boundary.

A.3 Generative Refinement Algorithm

We present the detailed algorithm of the proposed generative refinement with latent diffusion models in Sec. 3.4. In video diffusion model, a input video $\mathbf{V} \in \mathbb{R}^{T\times H\times W\times 3}$ is encoded to a latent vector via a encoder $\mathcal{E}$ by $\mathbf{z}_0 = \mathcal{E}(\mathbf{V}) \in \mathbb{R}^{T\times h\times w\times 3}$, where c is the dimension of the latent space. The forward diffusion process [35] is to iteratively add Gaussian noise to the signal, given by Eq. (1).

$$\begin{aligned} q\left(\mathbf{z}_t \mid \mathbf{z}_{t-1}\right) &= \mathcal{N}\left(\mathbf{z}_t; \sqrt{1-\beta_{t-1}}\mathbf{z}_{t-1}, \beta_t\mathbf{I}\right), \quad t = 1,\ldots,T \\ q\left(\mathbf{z}_t \mid \mathbf{z}_0\right) &= \mathcal{N}\left(\mathbf{z}_t; \sqrt{\bar{\alpha}_t}\mathbf{z}_0, (1-\bar{\alpha}_t)\mathbf{I}\right), \quad t = 1,\ldots,T \end{aligned} \tag{1}$$

where $\alpha_t = 1-\beta_t$, $\bar{\alpha}_t = \prod_{i=1}^{t}\alpha_i$, $q\left(\mathbf{z}_t \mid \mathbf{z}_{t-1}\right)$ is the one-step forward diffusion process, and $q\left(\mathbf{z}_t \mid \mathbf{z}_0\right)$ is t-step forward diffusion process. T is a large integer to make the forward process completely destroys the initial signal $\mathbf{z}_0$ resulting in $\mathbf{z}_T \sim \mathcal{N}(0,\mathbf{I})$. The diffusion model learns to recover $\mathbf{z}_0$ from standard Gaussian

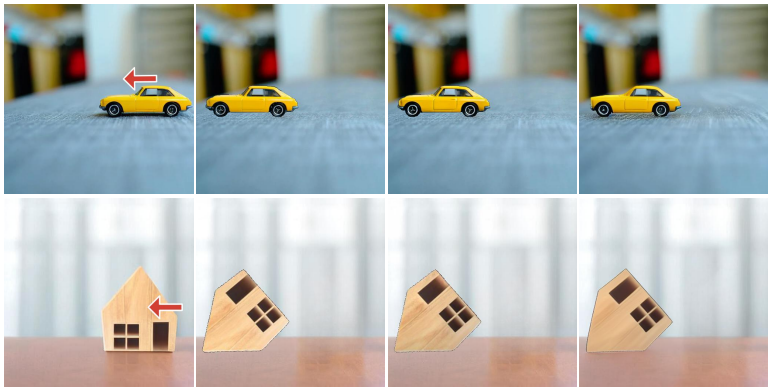

Fig. 4: Rendered Video comparison. The left shows the input frame, and the rest 3 are future frame generations. The composited frame hasn't been aware of the light change. The relighted output synthesized no shadows beneath. The rendered output from diffusion model is most photorealistic.

Table 1: Runtime analysis. We summarize the runtime of each module for each run.

Perception			Simulation	Render	Refinement
Segmentation	GPT4-V	Inpainting			
50s	10s	20s	5s	60s	35s

noise $\mathbf{z}_T$ by backward diffusion process in Eq. (2).

$$p_\theta\left(\mathbf{z}_{t-1} \mid \mathbf{z}_t\right)=\mathcal{N}\left(\mathbf{z}_{t-1} ; \mu_\theta\left(\mathbf{z}_t, t\right), \mathbf{\Sigma}_\theta\left(\mathbf{z}_t, t\right)\right), \quad t=T, \ldots, 1 \tag{2}$$

where θ is the learned parameters of the model. The output is passed to a decoder $\mathcal{D}$ to generate the output video $\mathbf{V}=\mathcal{D}(\mathbf{z}_0)$. To this end, given the relit video $\tilde{\mathbf{V}}$, we encode to get $\tilde{\mathbf{z}_0}$, our goal is to obtain the denoised $\mathbf{z}_0$ from the pretrained latent-diffusion-based video p_θ. The algorithm is shown in Algorithm 1.

In Fig. 4, we compare the composed video $\hat{\mathbf{V}}$, the relit video $\tilde{\mathbf{V}}$, and our final output $\mathbf{V}$ by 2 additional examples.

A.4 Runtime Analysis

We summarize the runtime of each module in Tab. 1 for a single run. The perception module takes 1 minute, the simulation (120 steps) takes 5 seconds, the render module takes 1 minute, the generative refinement takes 35 seconds. In total it takes around 3 minute for a single generation, which is much faster than other controllable generation, *e.g.* Motion Guidance [29] takes 70 minutes for a single run.

Algorithm 1 Video Generative Refinement

Input: Init $\tilde{\mathbf{z}_0}$, foreground mask m, video diffusion model p_θ, noise strength s, fusion timestamp δ, total denoising timesteps T
Output: refined latent $\mathbf{z}_0$

$T \leftarrow \lfloor T * s \rceil$ ▷ Denoised time steps
$\mathbf{z}_T, \tilde{\mathbf{z}_T} \sim q\left(\mathbf{z}_T \mid \tilde{\mathbf{z}_0}\right)$ ▷ Add noise to the guidance latent code
for $t = T, T-1, ..., 1$ **do**
  if $t \leq (T - \delta)$ **then**
    $\tilde{\mathbf{z}_{t-1}} \sim q\left(\mathbf{z}_{t-1} \mid \tilde{\mathbf{z}_0}\right)$ ▷ Add noise to guidance code at $t-1$
    $\mathbf{z}_{t-1} \sim p_\theta\left(\mathbf{z}_{t-1} \mid \mathbf{z}_t\right)$ ▷ Denoised output from network at $t-1$
    $w = (T - t)/T$ ▷ fusion weight
    $\mathbf{z}_{t-1} \leftarrow (1-m)\mathbf{z}_{t-1} + m\left[w\mathbf{z}_{t-1} + (1-w)\tilde{\mathbf{z}_{t-1}}\right]$ ▷ update latent code
  else
    $\mathbf{z}_{t-1} \sim p_\theta\left(\mathbf{z}_{t-1} \mid \mathbf{z}_t\right)$ ▷ Denoised output from network
  end if
end for
return $\mathbf{z}_0$

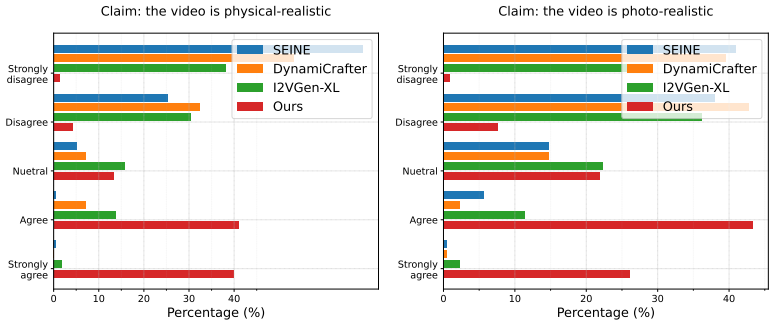

Fig. 5: Human evaluation score distribution. The distribution of scores shows our method largely outperforms other I2V generative models in both physical-realism and photo-realism. Our average rate is close to agree for both two claims.

B Experiment details

B.1 Human evaluation

Fig. 5 shows the score distribution of the conducted human evaluation. Our method has much higher percentage in agree and strongly-agree to both claims, outperform compared I2V methods by a large margin in physical-realism and photo-realism. Our average rate falls within Agree level.

B.2 Quantitative evaluation

To quantitative evaluate the generation performance and compare with other approaches, we record 50 videos for a given scene as GT by varying initial forces applied on the object. The random selected sequences are shown in Fig. 6.

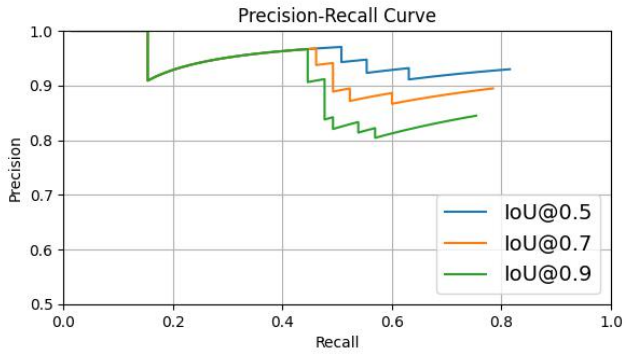

Fig. 7: Precision-recall curve of open-world movable objects segmentation. Our proposed pipeline achieves **0.93** precision, **0.82** recall at 0.5 IoU.

C More experiments

C.1 Open-world movable object segmentation

To measure the performance of the open-world movable object segmentation, we collect 10 images of very complicated open-world scenes where 118 movable instances are annotated following COCO [48] format. The perception system works well, achieves **0.93** precision **0.82** recall under 0.5 IoU. The precision-recall curve is shown in Fig. 7. Qualitative results are shown in Fig. 9.

C.2 GPT-4V physical property estimation evaluation

We evaluate the physical property estimator GPT-4V used in the paper. For *1) mass*: follow [95], we select 20 different portable objects from ABO dataset and find that GPT4-V has an average absolute error of **0.39 kg** and achieves **75%** accuracy within 30% of the GT. *2) friction and elasticity*: Since GT is unavailable, we use reference videos of toy cars sliding on various surfaces and balls of different materials bouncing to rank materials by friction and elasticity. Comparing the models' rankings to the ones in the videos, GPT-4V gives reasonable estimations, getting **12 out of 13** for friction and **6 out of 7** for elasticity across different comparisons. Two testing scenarios of friction and elasticity are shown in Fig. 8.

D More qualitative results

D.1 Qualitative comparison

We visualize more qualitative comparisons in Fig. 10 and Fig. 11. As can be seen, image-video generation methods could not generate physically plausible outputs. We notice DynamiCrafter [88] model tends to generate static scene with lighting

or viewpoint changes, without output reasonable physical motion. SEINE [19] and I2VGen-XL [96] outputs reasonably apparent motions, but the motions are not physical realistic, and the foreground objects could not guarantee consistency across different time steps. Please check the supplementary video for details.

D.2 More controllable generation

We show more controllable generation results in Fig. 12.

E Limitation Analysis

We summarize 4 different generation artifacts in Fig. 13.

Incorrect Inpainting. The first row shows the domino example. The generation results contains incorrect shadow in the middle of the frame. The reason is that the shadow of the boxes could not be fully removed in the inpainted image as shown on the right.

Gap between segmentations and primitives. The second row shows the blocks example where there is a gap between different blocks in the generated videos, whereas no such phenomenon appeared in simulation. The reason is the primitives used in the simulation is slight different from the real segments, thus could cause some gap in the simulated results and rendered outputs.

Inaccurate Segmentation. The third row shows the billiard example where the balls' shape is not perfect circle in the generated frames given the input segmentation mask is not accurate on the boundaries. Thus the synthesized ball has artifacts near its boundary. The generated video quality is affected by the segmentation mask of the input image.

Hallucinations introduced by diffusion refinement. The fourth rows shows the blocks example where the diffusion refinement brings hallucination of the input object. In our proposed generative refinement algorithm, the generated latent injects into both the foreground and background. Thus in some scenarios there could be hallucinations of the foreground object and slightly modify its appearance, *e.g.* the block on the top.

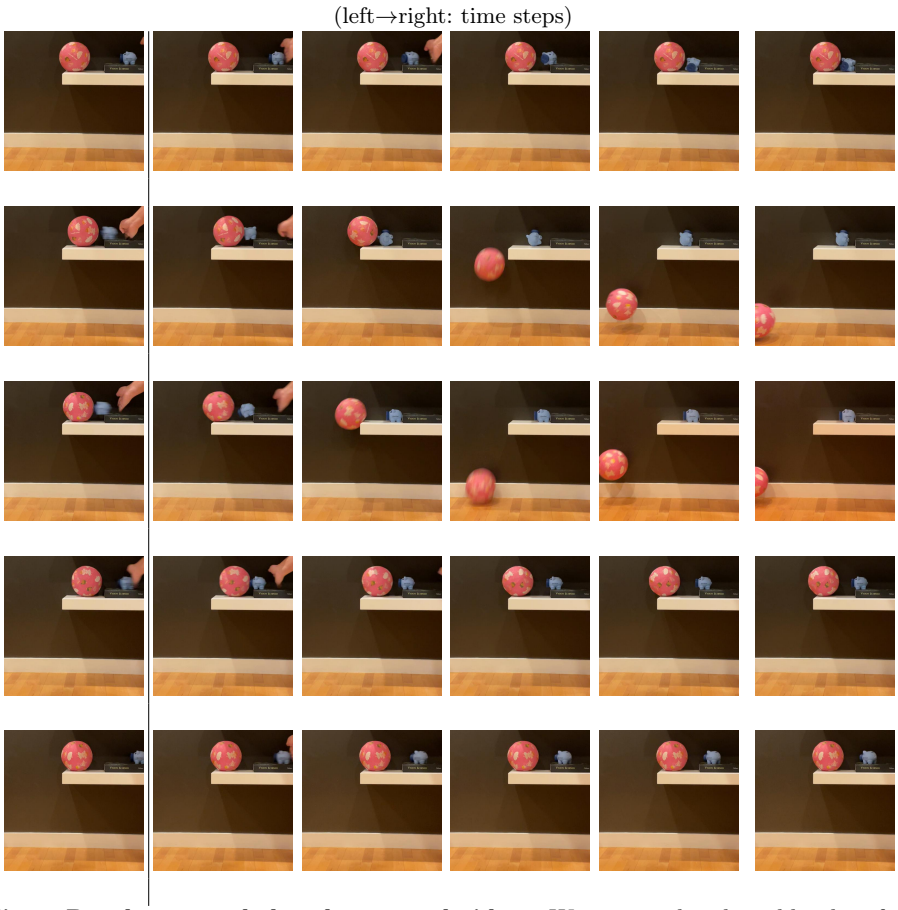

Fig. 6: Random sampled real-captured videos. We captured real-world videos for the same given scene 50 times to evaluate the generation fidelity. We vary initial force on the hand that applied to the blue piggy bank in each run, and record the videos. We **random select** 5 recored videos for visualization.

Fig. 8: GPT-4V physical property estimation evaluation testing scenes. Given directly measure friction and elasticity is hard, we use reference videos of toy cars sliding on various surfaces (left) and balls of different materials bouncing (right) to rank materials by friction and elasticity.

Fig. 9: Qualitative visualization of open-world movable objects segmentation results. From left to right, we show the selected input image, inferred segmentation from our perception module and GT movable object segmentations.

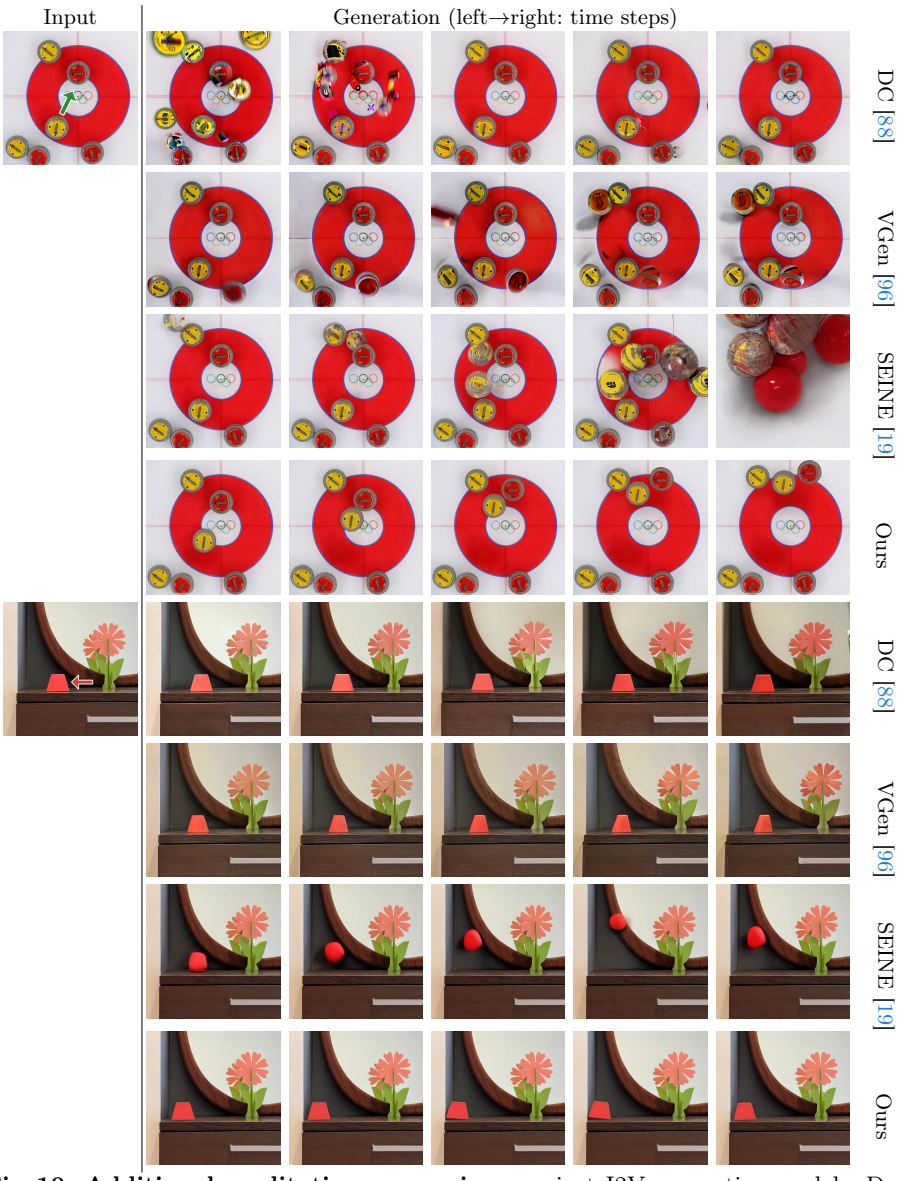

Fig. 10: Additional qualitative comparison against I2V generative models: DynamiCrafter(**DC**) [88], I2VGen-XL(**VGen**) [96], **SEINE** [19].

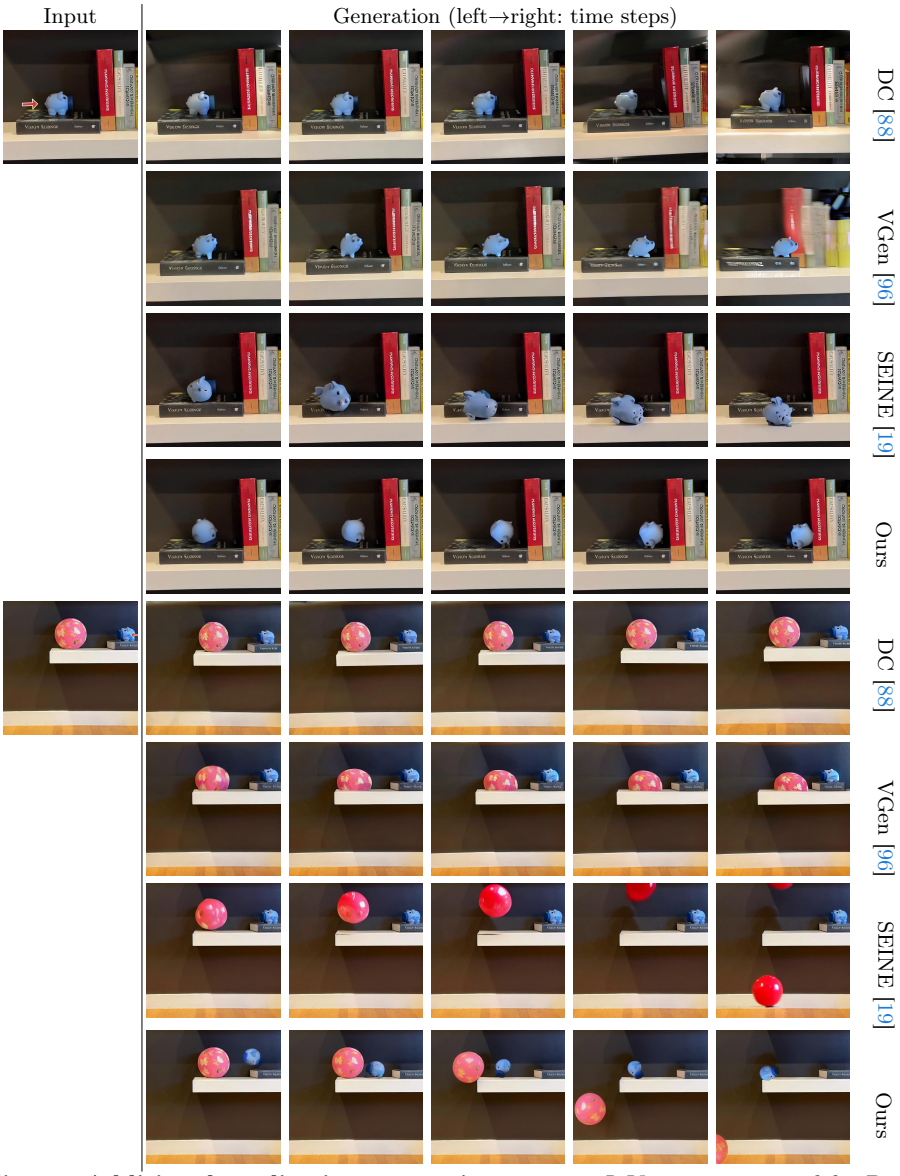

Fig. 11: Additional qualitative comparison against I2V generative models: DynamiCrafter(**DC**) [88], I2VGen-XL(**VGen**) [96], **SEINE** [19].

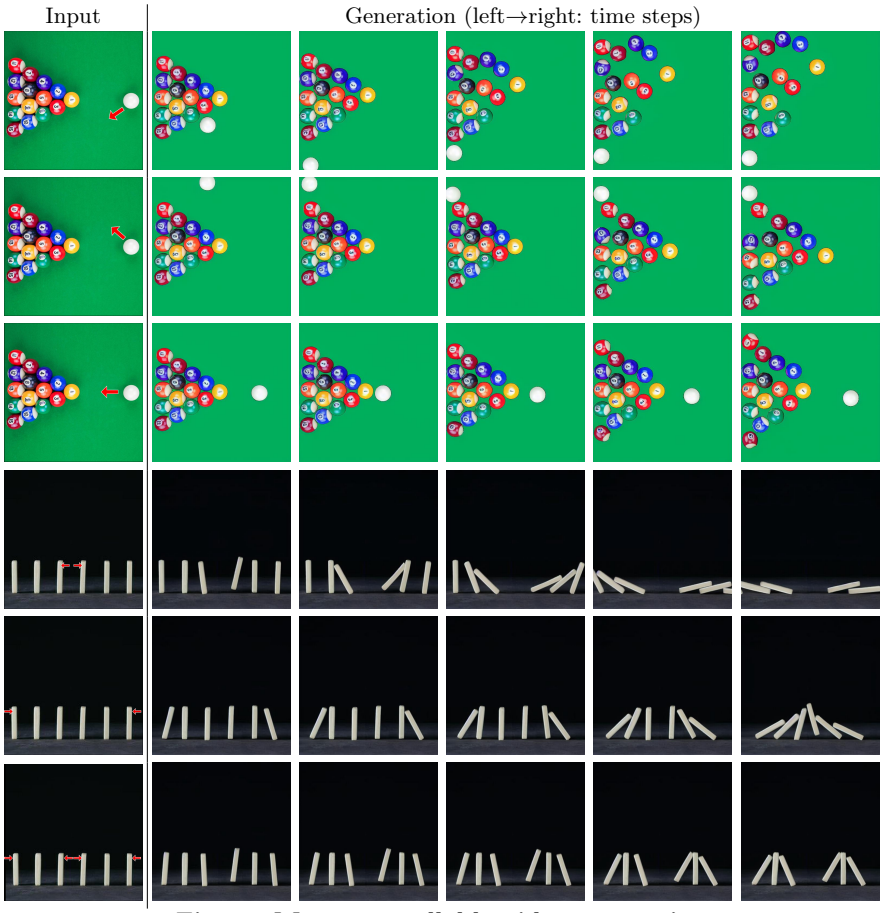

Fig. 12: More controllable video generation.

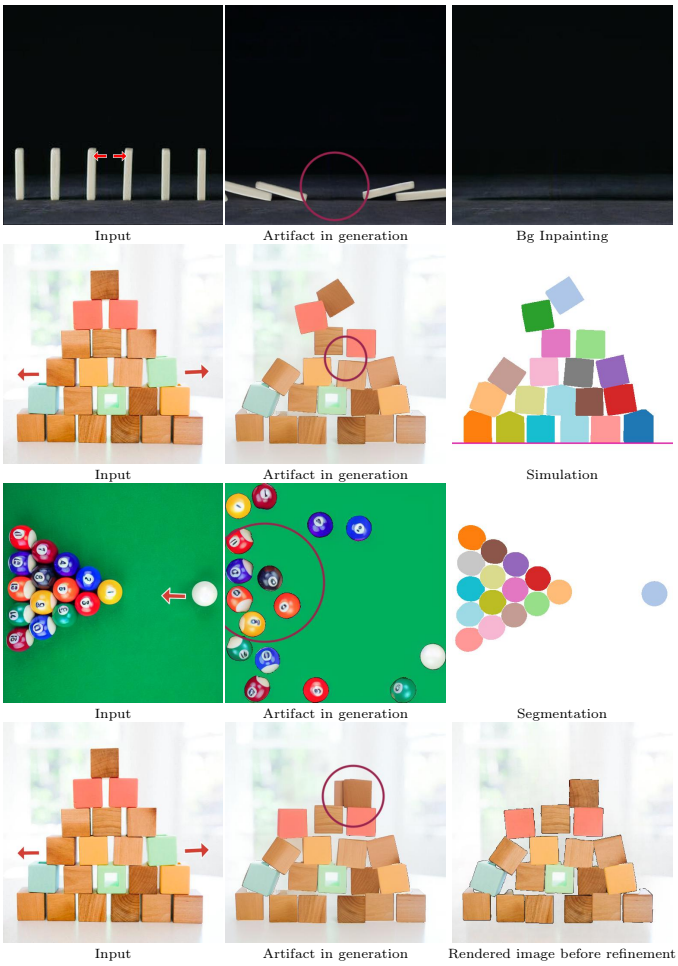

Fig. 13: Limitation analysis. We showcase 3 different examples of our current method's limitation. The left column shows the input image, the middle column shows the sampled frame from the generation video with artifacts, the right column shows the underlying reason for the artifact.