PhysGen3D: Crafting a Miniature Interactive World from a Single Image

Boyuan Chen[1], Hanxiao Jiang[2,3], Shaowei Liu[2],
Saurabh Gupta[2], Yunzhu Li[3], Hao Zhao[1], Shenlong Wang[2]
[1]Tsinghua University, [2]University of Illinois Urbana-Champaign, [3]Columbia University

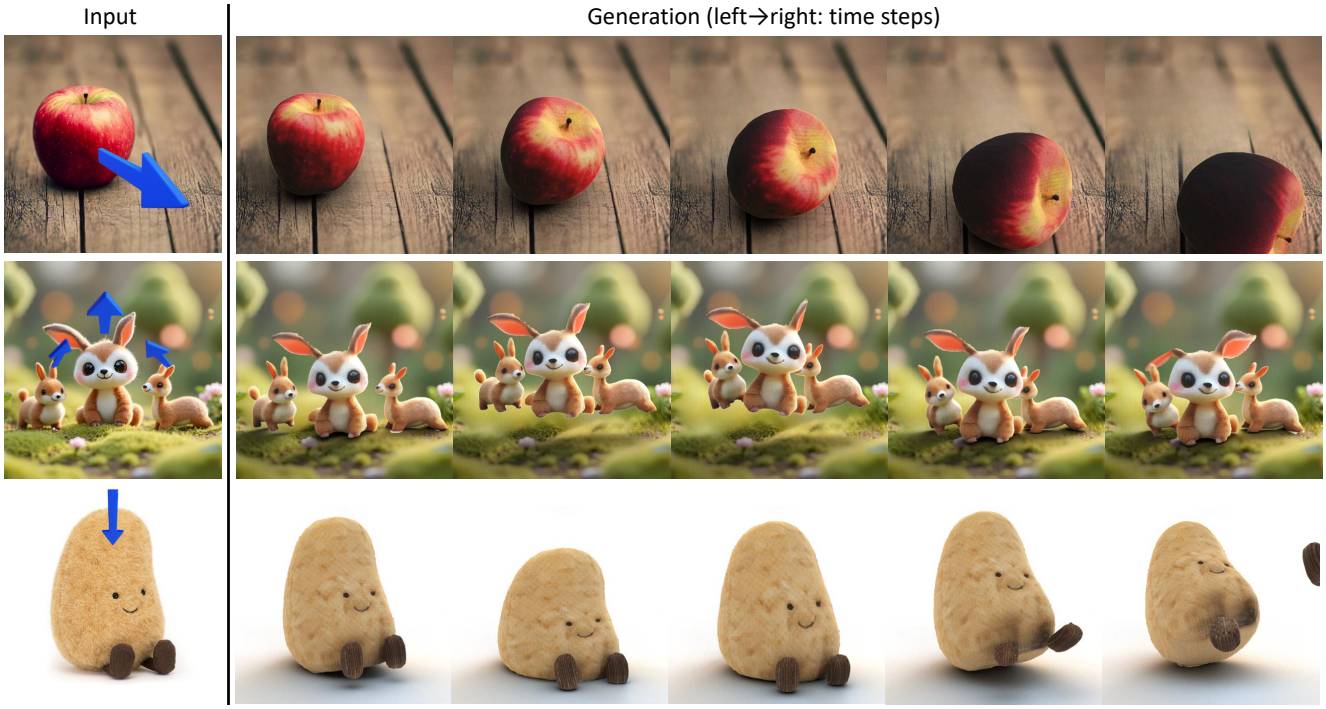

Figure 1. PhysGen3D generates realistic, physically plausible motion from a single image and a text prompt by reasoning about geometry, semantics, and material properties. (a) An apple rolls under the influence of its initial velocity, friction, and shape, producing a natural progression over time. (b) Three animal figures interact dynamically, colliding after being propelled upwards and forwards. (c) A toy potato bounces back with soft-body dynamics in response to an initial downward force, capturing material-specific behaviors. PhysGen3D lets users quickly explore physics-driven object interactions and behaviors in a compact virtual scene generated from a single input image.

Abstract

Envisioning physically plausible outcomes from a single image requires a deep understanding of the world's dynamics. To address this, we introduce PhysGen3D, a novel framework that transforms a single image into an amodal, camera-centric, interactive 3D scene. By combining advanced image-based geometric and semantic understanding with physics-based simulation, PhysGen3D creates an interactive 3D world from a static image, enabling us to "imagine" and simulate future scenarios based on user input. At its core, PhysGen3D estimates 3D shapes, poses, physical and lighting properties of objects, thereby capturing essential physical attributes that drive realistic object interactions. This framework allows users to specify precise initial conditions, such as object speed or material properties, for enhanced control over generated video outcomes. We evaluate PhysGen3D's performance against closed-source state-of-the-art (SOTA) image-to-video models, including Pika, Kling, and Gen-3, showing PhysGen3D's capacity to generate videos with realistic physics while offering greater flexibility and fine-grained control. Our results show that PhysGen3D achieves a unique balance of photorealism, physical plausibility, and user-driven interactivity, opening new possibilities for generating dynamic, physics-grounded video from an image. Project page: https://by-luckk.github.io/PhysGen3D.

1. Introduction

Photographs capture snapshots of our physical world, preserving specific moments in time but leaving out the alternative outcomes that could have unfolded. For instance, looking at a photo, we might wonder, “What if I poke the apples to make them roll across the ground?” or “What if I squeeze the three stuffed animals closer together?” or “What if I drop my cute potato toys onto the floor?” Humans intuitively understand how these scenarios would play out because we have an innate sense of the physical world beyond what we see in a single image. We develop a computational model that can answer such "what-if" questions by generating video outcomes from a single static image.

A promising approach toward this goal is data-driven image-to-video (I2V) generation [17, 18, 29, 63, 70, 77, 87]. I2V leverages diffusion-based generative models trained on vast datasets of internet images and videos, enabling the production of photorealistic videos with remarkable detail. However, I2V still has limitations in precise control and lacks physical grounding. As a result, users cannot interact freely and accurately to achieve specific physical effects, nor can I2V guarantee physical realism.

On the other hand, recent research has focused on modeling the physical world from visual inputs to create digital twins, allowing for precise interactions [43, 84, 96, 100]. These approaches can generate virtual scenes with convincing physical interactions, but they typically require complete 3D scans from multi-view images or depth sensors, making them data-intensive. While some methods [44, 49, 68] enable interaction with a single image, they are often limited by physical constraints (e.g., rigid bodies or springs), specific object types (e.g., waterfalls), or a 2D scope. This gap highlights the need for a generic, controllable, physically grounded, and photorealistic approach to generate video from a single image while maintaining physical realism.

In this work, we introduce PhysGen3D, a novel framework that transforms a single image into an amodal, camera-centric, interactive 3D scene, enabling realistic simulation and rendering. Our approach combines the strengths of image-based geometric and semantic understanding [40, 60, 69, 74, 86] with physics-based simulation [31–33]. At its core, PhysGen3D is a digital twinning method that estimates an object’s 3D shape, pose, physical and lighting properties, infers background geometry and appearance, deduces physical characteristics, and performs dimensional analysis—all from a single input image. This task is conventionally challenging due to its inherently ill-posed nature. To tackle this, we leverage various pretrained vision models, integrating their outputs to create an image-centric digital twin.

For physical simulation, we employ material point methods [30, 37], a robust point-voxel-based framework that models counterfactual physical behaviors of objects in the image. Through precise inference of physical properties, simulations in the PhysGen3D environment achieve a high degree of realism and stability. We further enhance realism by applying physics-based rendering, seamlessly integrating dynamic effects back into the original image. PhysGen3D produces results that are not only visually realistic in terms of dynamics and lighting but also highly controllable, allowing users to specify initial conditions like speed and material properties. Due to the use of large pretrained models, our pipeline operates effectively without task-specific training.

Our experiments, based on a carefully designed and rigorous user study, demonstrate that, compared to closed-source state-of-the-art video AIGC models such as Pika, Kling, and Gen-3, our framework provides significantly more flexible control over object motions, generates videos that better align with user intentions, and achieves superior physical realism—all while maintaining comparable rendering quality.

2. Related work

Single-view 3D reconstruction Great progress has been made in object-centric single view 3D reconstructions [46, 47, 64, 86]. However, applying these techniques to single-view 3D scene reconstruction becomes more challenging. Most existing works only focus on one part of 3D scene understanding. Geometric approaches reconstruct the 3D scene holistically but neglect individual object understanding, or they focus solely on foreground objects without considering the interaction of objects with the complex environment. Additionally, works have been done for scene relighting [50, 97, 99] and segmentation [40, 48, 60, 88, 101], but none of these provide a full 3D understanding within the image. In our work, we also perform perception reasoning for object materials, backgrounds, rendering properties, and physical stability.

Controllable video generation. Video generation has made significant progress in recent years [5, 9, 10, 15, 21, 23, 25, 28, 41, 56, 65, 72, 76, 85, 93, 95]. The state-of-the-art framework [1, 3] can generate photorealistic and coherent videos from text instructions using diffusion models [27, 57, 61, 66, 67]. Controllable video generation is achieved through leveraging pre-trained video generation models with conditioning information, including but not limited to depth maps [13, 89], linear translation [39], layouts [45, 73, 90], and multiple combinations [52]. However, most existing generation methods implicitly generate image space dynamics, which might lead to unrealistic, hallucinated motions. In contrast, our method explicitly controls the motion and interaction with simulation, allowing us to create more sophisticated effects without the need for extensive training data. Our approach is training-free and generalizable to all objects in the world.

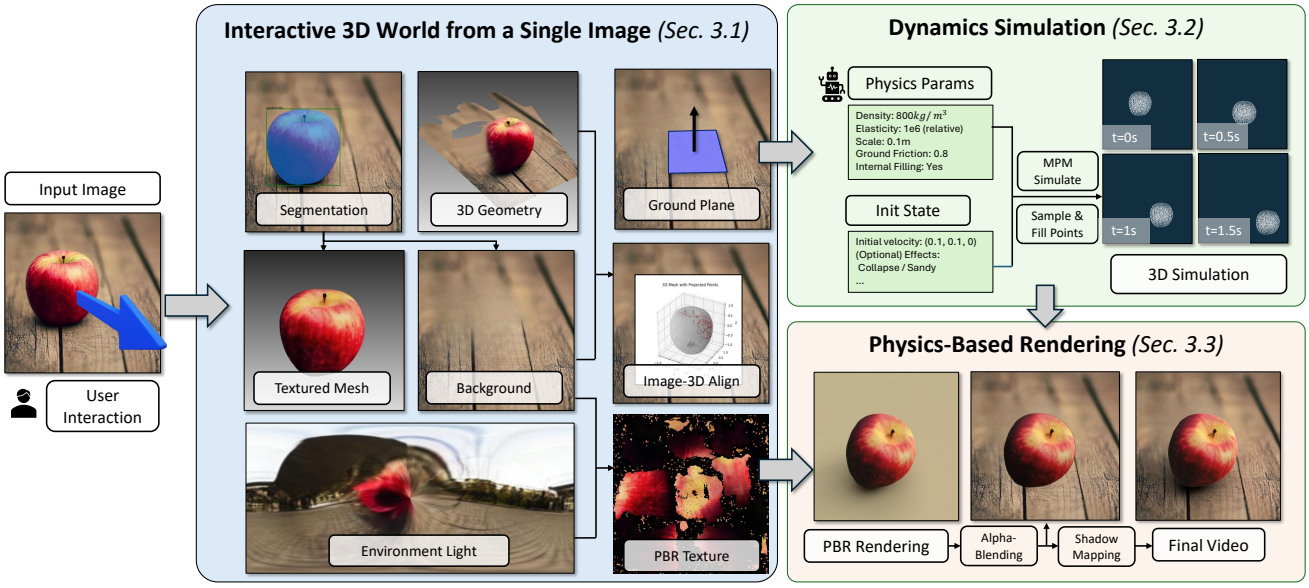

Figure 2. **Method Overview.** PhysGen3D's framework consists of three modules: a) 3D world creation, which infers geometry, semantics, rendering and physical parameters from the input image; b) dynamics simulation using Taichi-Elements for particle-based physics; and c) physics-based rendering with a two-pass shadow mapping technique.

Controllable image animation Image-based animation aims to animate objects that appear in images. Numerous works [17, 18, 29, 63, 70, 77, 87] have focused on this task. To improve quality, recent research has adopted data-driven solutions, training temporal neural networks to directly predict subsequent video frames [6, 14, 19, 24, 29, 75], or incorporating physical heuristics [17, 36]. Recently, there has been increased focus on interactive [7, 8, 44] and controllable [4, 18] image-to-video synthesis. Additional priors such as motion fields [19, 29, 53, 54, 68, 87], optical flow [11, 22, 98], 3D geometry information [59, 79], and user annotations [44] have also been introduced. The closest work to ours is Liu et al. [49], which focuses on image understanding and uses an existing 2D rigid simulator to generate 2D animations. However, their approach is limited to 2D dynamics, does not account for real-world 3D physics, and is restricted to rigid bodies. In contrast, our method provides more realistic and flexible control over 3D motions and object materials and extends beyond rigid-body physics.

3. Approach

Our goal is to reconstruct an interactive, camera-centric miniature world from a single input image. We aim to control object materials, dynamics, and motions to simulate diverse, photorealistic, and physically plausible videos from user input. The key challenge lies in partial observations and the ill-posed nature of physical reasoning without observed dynamics. To address this, we propose a holistic reconstruction method leveraging large pretrained visual models to jointly infer geometry, dynamic materials, lighting, and PBR materials from a single image (Sec. 3.1). The reconstructed scene is input into a material-point-method (MPM) simulator to generate realistic physics phenomena (Sec. 3.2). Finally, we render dynamic object behaviors based on the simulation and reintegrate them into the scene, producing realistic videos with accurate appearance and motion (Sec. 3.3). Fig. 2 depicts our framework.

3.1. Interactive 3D World from a Single Image

A full amodal reconstruction of the 3D world depicted in the image is critical for the next step in simulation. An ideal reconstruction should capture a comprehensive understanding of objects' relationships, geometry, appearance, material, and physical properties. However, obtaining this understanding from a single image is highly ill-posed. Our key idea is to leverage priors from pretrained vision foundation models to help infer these properties, as shown in Fig. 2.

Segmentation. We leverage vision foundation models to recognize object categories and segment object instances. Specifically, we use GPT-4o [92] to identify the foreground object categories and use Grounded-SAM [40, 48, 60] to further detect and segment each individual instance $\{o^i \in \mathbb{R}^{W\times H\times 3}\}_i^N$, where o^i is the image of i-th object.

Mesh Generation. Unlike previous work [49], which used only 2D rigid-body physics, we extend to general-purpose 3D simulation. This requires a complete 3D representation of foreground objects. We adopt InstantMesh [86], which uses Zero123++ [47, 64] to synthesize multi-view images from the segmented object image o^i and uses these images to reconstruct the 3D mesh $\mathcal{O}$ of the objects. For multiple object occlusions, we adopt the iterative inpainting

strategy to extract the 3D mesh of the objects (see supp for more details).

Background Handling. The background serves important roles in both dynamic simulation and rendering. In simulation, background geometry acts as a support and collider, where accuracy ensures realistic object-scene interactions. In rendering, it serves as a static backdrop while the foreground objects move and helps simulate realistic global illumination effects like cast shadows. For simulation purposes, we use Dust3r [74] to estimate iamge depth. The output depth map $z \in \mathbb{R}^{W \times H}$ is unprojected to the 3D world as a 3D point cloud $\mathcal{P}$, and Bilateral Normal Integration [12] is applied to generate a smooth surface $\mathcal{S}$ serving as the collider. For rendering, to fill background regions occluded by moving objects, we use the LaMA inpainting model[69]. This model generates a complete background after masking out all objects and their shadows.

Object Pose and Scale Estimation. The generative 3D reconstruction step provides a complete mesh with a normalized scale in an object-centric coordinate system, but it does not infer the object's location and scale in the camera coordinate. To ensure coherence with the input image, it is necessary to accurately place the 3D meshes $\mathcal{O}$ into the 3D scene $\mathcal{P}$ with the correct scale and 6DoF pose. This registration task is challenging, as the generated object mesh may not perfectly match the real-world object in the image.

To address this challenge, we designed a multi-stage coarse-to-fine alignment strategy. In the coarse stage, we perform 2D-3D feature point matching. Firstly, we render multiple images from viewpoints uniformly distributed over a unit sphere surrounding the object. For each rendered image, we match its feature points with the original object image o^i using SuperGlue [62], and the viewpoint with the most matches is selected. Matched pairs $({p'}_i^N \in \mathbb{R}^2, p_i^N \in \mathbb{R}^2)$ are then projected back into 3D points $({P'}_i^N \in \mathbb{R}^3, P_i^N \in \mathbb{R}^3)$ in object and camera coordinate, respectively. Perspective-n-Point (PnP) algorithm [20] is applied between ${P'}_i^N$ and p_i^N in image o^i to estimate the 6DoF pose with scale ambiguity. Then we adjust the scale and translation factor simultaneously to minimize the L2 loss $\sum_i^N = ||P_i^N - {P'}_i^N||_2$, without altering its projection.

In the fine alignment stage, we render the mask and depth on the image plane using the current estimation through a differentiable renderer and jointly minimize two losses: $\mathcal{L} = \mathcal{L}_{\text{dice}} + \mathcal{L}_{\text{depth}}$. Here, $\mathcal{L}_{\text{dice}} = 1 - \frac{2 \times |M_A \cap M_B|}{|M_A| + |M_B|}$ measures the discrepancy between the rendered mask M_A and the observed mask M_B from Grounded-SAM. The depth consistency loss is $\mathcal{L}_{\text{depth}} = \frac{\text{mse}(M_B * z_A - M_B * z_B)}{|M_B|}$, where z_A and z_B are the rendered and Dust3r-predicted depths. This joint optimization ensures consistency between the estimated mesh in the camera coordinate and the point cloud from Dust3r, while maximizing the alignment between the 3D pose and observed object mask, ensuring accurate simulation and rendering alignment.

Appearance Optimization. The texture of the generated 3D meshes may differ from the input image. To enhance rendering quality, we use the inverse rendering pipeline in Mitsuba3 [35] to estimate material properties. Lighting parameters are estimated with DiffusionLight [58], while object PBR materials (albedo, roughness, and metallic) are optimized via differentiable rendering. To handle unknown back views and reduce complexity, we assume uniform roughness and metallic values per object and apply tone mapping for albedo optimization: $y(x) = ax^3 + bx^2 + cx + d, \quad \text{where } y(0) = 0, y(1) = 1$. With reconstructed lighting and ground surface, the optimized materials improve asset appearance, capturing realistic object-surface interactions during rendering.

Physics Reasoning. Accurately simulating real-world dynamics requires estimating physical parameters. We focus on two aspects: 1) Following [82], we use GPT-4o to query each object's elasticity and density, and the friction coefficient for the surface $\mathcal{S}$. 2) We ensure reconstructed 3D assets match real-world proportions, as depth inaccuracies from Dust3r can cause unrealistic behaviors. To address this, we estimate a scale factor k by comparing asset size with typical real-world sizes from GPT-4o and use k for dimensionless scaling of gravity and velocity-related parameters.

3.2. Dynamics Simulation

Given the 3D assets with reasoned physical properties and the scale factor, we use the physics engine Taichi Elements [31–33], based on the Material Point Method (MPM) [37], as our simulator to support a variety of different materials, including but not limited to rigid, soft, and granular.

Particle representation. The simulator is based on a particle-based representation. To convert our 3D assets into a simulatable particle-based representation, we apply floating points removing, internal filling and voxel downsampling. We use downsampling to handle uneven particle distributions, where downsample rate is adjusted according to the grid size of the simulator. For the convenience of rendering, we prioritize surface points.

Physical parameters. To enhance stability, we apply the scale factor k to physical parameters in simulator instead of scaling the assets to real size. For example, the gravitational acceleration is multiplied by k, making to $k * 9.8$ (nondimensionalized here). With this tuning, the motion of falling or collapse remains realistic for all scales.

External disturbance. For each object, we set a different initial velocity based on the user input to make the object move as specified by the user.

Other Visual Effects. Besides real world physics simulation, our pipeline allows special effects like collapsing and melting. To simulate different materials (rigid, soft, or granular), we can easily change the material type to modify

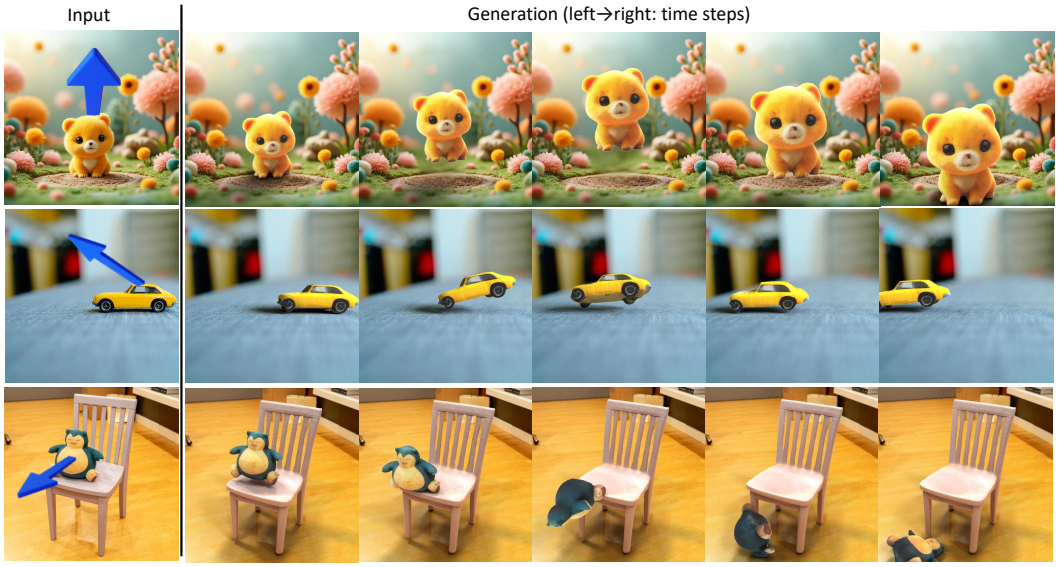

Figure 3. **Video generation results.** Left: Input initial frame. Right: Generated future frames. We apply an initial velocity to each movable object and use the physically grounded parameters outlined in 3.2 to generate physically plausible results.

the physical properties of an object. This approach provides more flexibility for the user to edit the scene.

3.3. Physics-Based Rendering

After dynamics simulation, we obtain object point trajectories and apply motion interpolation to compute vertex motions, deforming the mesh accordingly. With optimized PBR materials, we use Mitsuba3 for Physically-Based Rendering under environment lighting. Following prior insertion rendering work [16, 38, 78, 91], we avoid converting the entire static background into the rendering pipeline. Instead, we build a 3D shadow catcher surface from the background depth. During rendering, no texture is applied to the background; two-pass shadow mapping extracts shadows and global illumination effects. The foreground objects and shadows are then composited onto the inpainted background to produce the final video with realistic lighting.

4. Experimental Results

4.1. Setups

The test images come from a diverse set of our own photography, online collections, and generative models. Our pipeline is primarily designed for object-centric scenes with one or a few objects. Due to our limitations (Sec. 5), we excluded those with excessive objects, heavy occlusion between objects, or highly uneven surfaces.

Post-processing. We used VEnhancer [26] as an optional post-processing module, which takes the produced video and a text prompt to perform enhancement. As shown in Table 2, while it restores some details, it also introduces extra hallucinations.

Baselines. Our method is one of the first of its kind, as existing model-simulate approaches require multi-view images [81, 83, 96] or specific settings [44, 49, 68]. Therefore, we evaluate ours against Image-to-Video (I2V) models: two open-source motion controller models DragAnything[80], MOFA-Video[55] and three state-of-the-art (SOTA) commercial models Kling 1.0 [3], Gen-3 [2], and Pika 1.5 [1]. We manually set correct motion trajectories and select applied regions for DragAnything, MOFA-Video and Kling to provide privileged motion guidance. We give text descriptions to Pika 1.5 and Gen-3, as they lack direct motion control capabilities. Additionally, Pika 1.5 offers "Pikaffect" effects, such as "Melt it" and "Deflate it."

4.2. Results

Our system generates a miniature interactive world from a single image, enabling the simulation of various phenomena. Fig. 3 presents videos generated from different types of images. These images encompass single or multiple objects and various physical materials (i.e., rigid or soft). We also show applications like dynamic changes, object editing, and dense 3D tracking, illustrating the adaptability and creative potential of our approach for generating customized, interactive video content.

Comparisons. We compare the results in two dimensions: motion control alone and physical materials. Fig. 4 shows that our system produces more physically plausible and controllable videos compared to SOTA I2V models. Despite prompt tuning, learning-based models often hallucinate, failing to adhere to physical laws or user intent. For example, in the toy dog case, we manually adjust the material and accurately simulate a collapse, whereas other models fall short. Similarly, for the book case, our results are the most physically realistic.

Dynamics. Fig. 5 demonstrates the varying dynamics generated from the same input image, highlighting the high

Figure 4. **Qualitative comparison.** We compare videos generated from our framework with three state-of-the-art I2V models: Gen-3 [2], Pika [1] and Kling [3]. We carefully designed the prompt to describe the motion outcome, and uses motion brush to control Kling. Our framework employs initial velocity control. Results show that our method can follow text instructions while maintaining plausible physics.

controllability of our method over physical parameters and motion trajectories. In the three rows on the left, we set different elasticities for the two objects, while keeping their initial positions and velocities the same. In the three rows on the right, we alter velocity directions of the objects and keep physical parameters the same.

Editing. Our method enables modifications to videos by removing, adding, or replacing objects, as illustrated in Fig. 6. The generated 3D assets can be easily manipulated, allowing for diverse video edits.

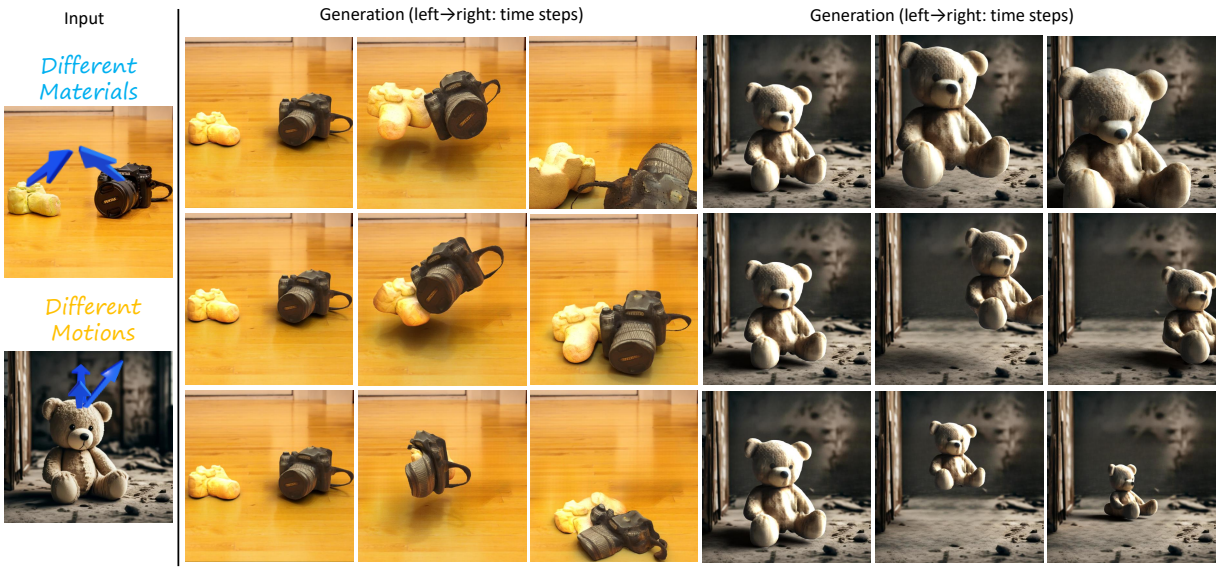

Figure 5. **Dynamics Effects.** We can generate various dynamics from the same input image. The left three columns share the same initial positions and velocities, but are in different materials. The right three columns have the same material, but defer in velocity directions. This showcases the potential of our method for generating diverse physical scenarios.

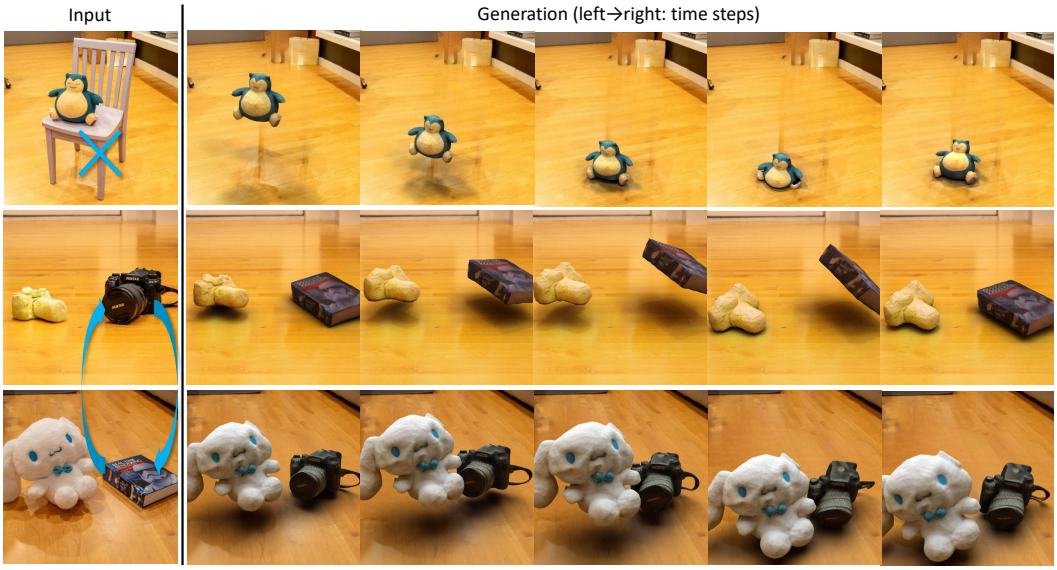

Figure 6. **Video edition.** As illustrated, our method supports video editing. In the top row, we remove the chair and allow the toy to fall from a static position. In the bottom two rows, we exchange one object between two scenes while keeping the other unchanged. This demonstrates the great flexibility of our video generation approach.

Tracking. Our framework uses an explicit 3D representation and works with a particle-based physics simulator. This allows our method to easily create videos with detailed 3D tracking results. Fig. 7 showcases two examples, demonstrating the accuracy and reliability of the tracking in different scenarios.

Ablation. Each step we designed in perception, simulation and rendering is intended to mimic the real world. Fig. 8 shows the results of ablation study involving position optimization and inverse texture. Without pose optimization using differentiable rendering, the two objects are roughly at the initial place shown in input image, but cannot fully replicate the scene. Without inverse texture, the generated object mesh does not match the input in terms of color tone and brightness. We also conducted ablations on point sampling. However, without sampling, a large number of points become crowded in several MPM simulation grids, causing the simulator to crash and fail to produce final outputs.

Figure 7. **Dense 3D Tracking.** Our method can naturally generate videos with dense 3D tracking results. Here we show tracking results for collapse and bounce cases.

Figure 8. **Ablation study.** Without pose optimization, the two objects cannot fully replicate the scene. Without inverse texture, the object mesh does not match the input color.

Table 1. **Human Evaluation Results.** The three criteria are: Physical Realism, Photorealism and Semantic Consistency

Methods	PhysReal	PhotoReal	Align
Kling 1.0 [3]	2.811	3.566	2.467
Runway Gen-3 [2]	2.283	**3.582**	1.886
Pika 1.5 [1]	2.412	3.314	2.016
Ours	**3.707**	3.411	**3.866**

Table 2. **VBench Scores and GPT-4o Evaluation Results. Motion** and **Imaging** refers to Motion Smoothness and Imaging Quality scores in VBench. The three criteria on the right are the same as Table 1, with results given by GPT-4o.

Methods	Motion↑	Imaging↑	PhysReal↑	PhotoReal↑	Align↑
Kling 1.0	**0.996**	0.671	0.563	0.874	0.596
Runway Gen-3	0.991	**0.723**	0.141	**0.896**	0.144
Pika 1.5	0.994	0.671	0.544	0.863	0.563
MOFA-Video	0.994	0.634	0.384	0.764	0.304
DragAnything	0.985	0.428	0.645	0.756	0.380
Ours	0.995	0.666	0.752	0.867	**0.796**
Ours+VEnhancer	0.994	0.677	**0.766**	0.880	0.745

4.3. Quantitative Comparison

To assess the quality of the generated videos, we performed human evaluation, GPT-4o evaluation and provided two VBench[34] evaluation scores.

Benchmarks. We designed three criteria for human and GPT-4o evaluation. (1)**Physical Realism (PhysReal)** measures how realistically the video follows the physical rules and whether the video represents real physical properties like elasticity and friction. (2) **Photorealism (Photoreal)** assesses the overall visual quality of the video, including the visual artifacts, discontinuities, and how accurately the video replicates details of light, shadow, texture, and materials. (3) **Semantic Consistency (Align)** evaluates how well the content of the generated video aligns with the intended meaning of the text prompt. We also chose two Quality Scores in VBench: Motion Smoothness and Imaging Quality.

Details in evaluation. Following a similar methodology to [49], we designed a questionnaire with 27 videos covering various scenes, motion conditions, and effects. Each video includes an input image, a motion prompt, and outputs from three SOTA commercial baselines and ours, shown in random order. 31 participants rated three criteria on a five-point scale from strongly disagree (1) to strongly agree (5). We also evaluated all five baselines and our diffusion-enhanced version using GPT-4o and VBench. GPT-4o assessed videos on the same criteria based on 10 evenly sampled frames, with the input image and prompt provided. We used GPT version gpt-4o-2024-08-06, with detailed prompts in the supp.

Result analysis. The results in Table 1 demonstrate our ability to generate physically accurate and controllable videos. In physical realism (**PhysReal**) and semantic consistency (**Align**), our method achieves the highest scores and outperform all the commercial models by a large margin. Table 2 presents the results of GPT-4o and VBench evaluation. GPT score is aligned with human evaluation, and we also beat open-sourced in VBench. Among the baseline models, Kling 1.0 performs the best overall, likely due to its use of the "motion brush", which specifies motion trajectories.

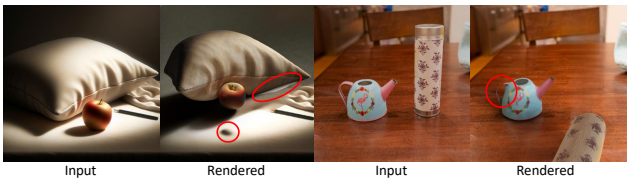

Figure 9. **Limitations.** The first set of two images show rendering errors: extreme lighting and heavy shading cause incorrect painting and artifacts from the pillow penetrating the ground. The second set reveals simulation limitations: the MPM method struggles with fine details, such as the thin kettle handle, leading to failures.

5. Limitations

Our single-image-based interactive miniature world is designed for object-centric scenes with simple spatial geometry and inter-object relationships. Reconstructing complete scenes for more complex scenarios remains an open challenge. Fig. 9 illustrates several failure cases, such as rendering errors under challenging shading, perception failures, and simulation limitations. More detailed analysis on limitations can be found in the supp.

6. Conclusions

We present PhysGen3D, a framework that transforms a static image into an interactive 3D scene for simulating and rendering future motions based on user input. PhysGen3D integrates modules for 3D world reconstruction, model-based dynamic simulation, and physics-based rendering to generate realistic, controllable videos. By extending the 2D image-to-video paradigm to 3D, PhysGen3D enables more realistic motion and diverse material behaviors. We hope this work inspires future research.

Acknowledgement

This project is supported by the Intel AI SRS gift, Amazon-Illinois AICE grant, Meta Research Grant, IBM IIDAI Grant, and NSF Awards #2331878, #2340254, #2312102, #2414227, and #2404385. We greatly appreciate the NCSA for providing computing resources.

References

[1] Pika, 2024. 2, 5, 6, 8

[2] Runway, 2024. 5, 6, 8

[3] Kling ai, 2024. 2, 5, 6, 8

[4] Kfir Aberman, Yijia Weng, Dani Lischinski, Daniel Cohen-Or, and Baoquan Chen. Unpaired motion style transfer from video to animation. *ACM TOG*, 2020. 3

[5] Omer Bar-Tal, Hila Chefer, Omer Tov, Charles Herrmann, Roni Paiss, Shiran Zada, Ariel Ephrat, Junhwa Hur, Yuanzhen Li, Tomer Michaeli, et al. Lumiere: A space-time diffusion model for video generation. *arXiv preprint arXiv:2401.12945*, 2024. 2

[6] Andreas Blattmann, Timo Milbich, Michael Dorkenwald, and Bjorn Ommer. Understanding object dynamics for interactive image-to-video synthesis. In *CVPR*, 2021. 3

[7] Andreas Blattmann, Timo Milbich, Michael Dorkenwald, and Björn Ommer. ipoke: Poking a still image for controlled stochastic video synthesis. In *ICCV*, 2021. 3

[8] Andreas Blattmann, Timo Milbich, Michael Dorkenwald, and Bjorn Ommer. Understanding object dynamics for interactive image-to-video synthesis. In *CVPR*, 2021. 3

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

[10] Andreas Blattmann, Robin Rombach, Huan Ling, Tim Dockhorn, Seung Wook Kim, Sanja Fidler, and Karsten Kreis. Align your latents: High-resolution video synthesis with latent diffusion models. In *CVPR*, 2023. 2

[11] Richard Strong Bowen, Richard Tucker, Ramin Zabih, and Noah Snavely. Dimensions of motion: Monocular prediction through flow subspaces. In *3DV*, 2022. 3

[12] Xu Cao, Hiroaki Santo, Boxin Shi, Fumio Okura, and Yasuyuki Matsushita. Bilateral normal integration. In *European Conference on Computer Vision*, pages 552–567. Springer, 2022. 4

[13] Weifeng Chen, Yatai Ji, Jie Wu, Hefeng Wu, Pan Xie, Jiashi Li, Xin Xia, Xuefeng Xiao, and Liang Lin. Control-a-video: Controllable text-to-video generation with diffusion models. *arXiv preprint arXiv:2305.13840*, 2023. 2

[14] Xi Chen, Zhiheng Liu, Mengting Chen, Yutong Feng, Yu Liu, Yujun Shen, and Hengshuang Zhao. Livephoto: Real image animation with text-guided motion control. *arXiv preprint arXiv:2312.02928*, 2023. 3

[15] Xinyuan Chen, Yaohui Wang, Lingjun Zhang, Shaobin Zhuang, Xin Ma, Jiashuo Yu, Yali Wang, Dahua Lin, Yu Qiao, and Ziwei Liu. Seine: Short-to-long video diffusion model for generative transition and prediction. *arXiv preprint arXiv:2310.20700*, 2023. 2

[16] Yun Chen, Frieda Rong, Shivam Duggal, Shenlong Wang, Xinchen Yan, Sivabalan Manivasagam, Shangjie Xue, Ersin Yumer, and Raquel Urtasun. Geosim: Realistic video simulation via geometry-aware composition for self-driving. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 7230–7240, 2021. 5

[17] Yung-Yu Chuang, Dan B Goldman, Ke Colin Zheng, Brian Curless, David H Salesin, and Richard Szeliski. Animating pictures with stochastic motion textures. *ACM TOG*, 2005. 2, 3

[18] Abe Davis, Justin G Chen, and Frédo Durand. Image-space modal bases for plausible manipulation of objects in video. *ACM TOG*, 2015. 2, 3

[19] Yuki Endo, Yoshihiro Kanamori, and Shigeru Kuriyama. Animating landscape: self-supervised learning of decoupled motion and appearance for single-image video synthesis. *arXiv preprint arXiv:1910.07192*, 2019. 3

[20] Martin A Fischler and Robert C Bolles. Random sample consensus: a paradigm for model fitting with applications to image analysis and automated cartography. *Communications of the ACM*, 24(6):381–395, 1981. 4

[21] Songwei Ge, Seungjun Nah, Guilin Liu, Tyler Poon, Andrew Tao, Bryan Catanzaro, David Jacobs, Jia-Bin Huang, Ming-Yu Liu, and Yogesh Balaji. Preserve your own correlation: A noise prior for video diffusion models. In *ICCV*, 2023. 2

[22] Daniel Geng and Andrew Owens. Motion guidance: Diffusion-based image editing with differentiable motion estimators. In *ICLR*, 2024. 3

[23] Rohit Girdhar, Mannat Singh, Andrew Brown, Quentin Duval, Samaneh Azadi, Sai Saketh Rambhatla, Akbar Shah, Xi Yin, Devi Parikh, and Ishan Misra. Emu video: Factorizing text-to-video generation by explicit image conditioning. *arXiv preprint arXiv:2311.10709*, 2023. 2

[24] 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, 2023. 3

[25] Agrim Gupta, Lijun Yu, Kihyuk Sohn, Xiuye Gu, Meera Hahn, Li Fei-Fei, Irfan Essa, Lu Jiang, and José Lezama. Photorealistic video generation with diffusion models. *arXiv preprint arXiv:2312.06662*, 2023. 2

[26] Jingwen He, Tianfan Xue, Dongyang Liu, Xinqi Lin, Peng Gao, Dahua Lin, Yu Qiao, Wanli Ouyang, and Ziwei Liu. Venhancer: Generative space-time enhancement for video generation. *arXiv preprint arXiv:2407.07667*, 2024. 5

[27] Jonathan Ho, Ajay Jain, and Pieter Abbeel. Denoising diffusion probabilistic models. 2020. 2

[28] 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. 2

[29] Aleksander Holynski, Brian L Curless, Steven M Seitz, and Richard Szeliski. Animating pictures with eulerian motion fields. In *CVPR*, 2021. 2, 3

[30] Yuanming Hu, Yu Fang, Ziheng Ge, Ziyin Qu, Yixin Zhu, Andre Pradhana, and Chenfanfu Jiang. A moving least

squares material point method with displacement discontinuity and two-way rigid body coupling. *ACM TOG*, 2018. 2

[31] Yuanming Hu, Tzu-Mao Li, Luke Anderson, Jonathan Ragan-Kelley, and Frédo Durand. Taichi: a language for high-performance computation on spatially sparse data structures. *ACM Transactions on Graphics (TOG)*, 38(6):201, 2019. 2, 4, 14

[32] Yuanming Hu, Luke Anderson, Tzu-Mao Li, Qi Sun, Nathan Carr, Jonathan Ragan-Kelley, and Frédo Durand. Difftaichi: Differentiable programming for physical simulation. *ICLR*, 2020.

[33] Yuanming Hu, Jiafeng Liu, Xuanda Yang, Mingkuan Xu, Ye Kuang, Weiwei Xu, Qiang Dai, William T. Freeman, and Frédo Durand. Quantaichi: A compiler for quantized simulations. *ACM Transactions on Graphics (TOG)*, 40(4), 2021. 2, 4, 14

[34] Ziqi Huang, Yinan He, Jiashuo Yu, Fan Zhang, Chenyang Si, Yuming Jiang, Yuanhan Zhang, Tianxing Wu, Qingyang Jin, Nattapol Chanpaisit, Yaohui Wang, Xinyuan Chen, Limin Wang, Dahua Lin, Yu Qiao, and Ziwei Liu. VBench: Comprehensive benchmark suite for video generative models. In *CVPR*, 2024. 8

[35] Wenzel Jakob, Sébastien Speierer, Nicolas Roussel, and Delio Vicini. Dr.jit: A just-in-time compiler for differentiable rendering. *Transactions on Graphics (Proceedings of SIGGRAPH)*, 41(4), 2022. 4

[36] Wei-Cih Jhou and Wen-Huang Cheng. Animating still landscape photographs through cloud motion creation. *IEEE Transactions on Multimedia*, 2015. 3

[37] Chenfanfu Jiang, Craig Schroeder, Joseph Teran, Alexey Stomakhin, and Andrew Selle. The material point method for simulating continuum materials. In *Acm siggraph 2016 courses*, pages 1–52. 2016. 2, 4

[38] Kevin Karsch, Varsha Hedau, David Forsyth, and Derek Hoiem. Rendering synthetic objects into legacy photographs. *ACM Transactions on graphics (TOG)*, 30(6):1–12, 2011. 5

[39] Levon Khachatryan, Andranik Movsisyan, Vahram Tadevosyan, Roberto Henschel, Zhangyang Wang, Shant Navasardyan, and Humphrey Shi. Text2video-zero: Text-to-image diffusion models are zero-shot video generators. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 15954–15964, 2023. 2

[40] Alexander Kirillov, Eric Mintun, Nikhila Ravi, Hanzi Mao, Chloe Rolland, Laura Gustafson, Tete Xiao, Spencer Whitehead, Alexander C. Berg, Wan-Yen Lo, Piotr Dollár, and Ross Girshick. Segment anything. *arXiv:2304.02643*, 2023. 2, 3

[41] Dan Kondratyuk, Lijun Yu, Xiuye Gu, José Lezama, Jonathan Huang, Rachel Hornung, Hartwig Adam, Hassan Akbari, Yair Alon, Vighnesh Birodkar, et al. Videopoet: A large language model for zero-shot video generation. *arXiv preprint arXiv:2312.14125*, 2023. 2

[42] Chenghua Li, Bo Yang, Zhiqi Wu, Gao Chen, Yihan Yu, and Shengxiao Zhou. Shadow removal based on diffusion segmentation and super-resolution models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 6045–6054, 2024. 13

[43] Xuan Li, Yi-Ling Qiao, Peter Yichen Chen, Krishna Murthy Jatavallabhula, Ming Lin, Chenfanfu Jiang, and Chuang Gan. Pac-nerf: Physics augmented continuum neural radiance fields for geometry-agnostic system identification. In *ICLR*, 2023. 2

[44] Zhengqi Li, Richard Tucker, Noah Snavely, and Aleksander Holynski. Generative image dynamics. *arXiv preprint arXiv:2309.07906*, 2023. 2, 3, 5

[45] Long Lian, Baifeng Shi, Adam Yala, Trevor Darrell, and Boyi Li. Llm-grounded video diffusion models. *arXiv preprint arXiv:2309.17444*, 2023. 2

[46] Minghua Liu, Ruoxi Shi, Linghao Chen, Zhuoyang Zhang, Chao Xu, Xinyue Wei, Hansheng Chen, Chong Zeng, Jiayuan Gu, and Hao Su. One-2-3-45++: Fast single image to 3d objects with consistent multi-view generation and 3d diffusion. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 10072–10083, 2024. 2

[47] Ruoshi Liu, Rundi Wu, Basile Van Hoorick, Pavel Tokmakov, Sergey Zakharov, and Carl Vondrick. Zero-1-to-3: Zero-shot one image to 3d object, 2023. 2, 3

[48] Shilong Liu, Zhaoyang Zeng, Tianhe Ren, Feng Li, Hao Zhang, Jie Yang, Chunyuan Li, Jianwei Yang, Hang Su, Jun Zhu, et al. Grounding dino: Marrying dino with grounded pre-training for open-set object detection. *arXiv preprint arXiv:2303.05499*, 2023. 2, 3

[49] Shaowei Liu, Zhongzheng Ren, Saurabh Gupta, and Shenlong Wang. Physgen: Rigid-body physics-grounded image-to-video generation. In *European Conference on Computer Vision*, pages 360–378. Springer, 2025. 2, 3, 5, 8

[50] Xinyu Liu, Jinlong Li, Jin Ma, Huiming Sun, Zhigang Xu, Tianyun Zhang, and Hongkai Yu. Deep transfer learning for intelligent vehicle perception: A survey. *Green Energy and Intelligent Transportation*, page 100125, 2023. 2

[51] Zhihao Liu, Hui Yin, Xinyi Wu, Zhenyao Wu, Yang Mi, and Song Wang. From shadow generation to shadow removal. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 4927–4936, 2021. 13

[52] Wan-Duo Kurt Ma, John P Lewis, and W Bastiaan Kleijn. Trailblazer: Trajectory control for diffusion-based video generation. *arXiv preprint arXiv:2401.00896*, 2023. 2

[53] Aniruddha Mahapatra and Kuldeep Kulkarni. Controllable animation of fluid elements in still images. In *CVPR*, 2022. 3

[54] Arun Mallya, Ting-Chun Wang, and Ming-Yu Liu. Implicit warping for animation with image sets. 2022. 3

[55] 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 *European Conference on Computer Vision*, pages 111–128. Springer, 2024. 5

[56] OpenAI. Creating video from text. https://openai.com/sora, 2024. 2

[57] William Peebles and Saining Xie. Scalable diffusion models with transformers. *arXiv preprint arXiv:2212.09748*, 2022. 2

[58] Pakkapon Phongthawee, Worameth Chinchuthakun, Nontaphat Sinsunthithet, Varun Jampani, Amit Raj, Pramook Khungurn, and Supasorn Suwajanakorn. Diffusionlight: Light probes for free by painting a chrome ball. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 98–108, 2024. 4

[59] Haonan Qiu, Zhaoxi Chen, Yuming Jiang, Hang Zhou, Xiangyu Fan, Lei Yang, Wayne Wu, and Ziwei Liu. Relitalk: Relightable talking portrait generation from a single video, 2023. 3

[60] Tianhe Ren, Shilong Liu, Ailing Zeng, Jing Lin, Kunchang Li, He Cao, Jiayu Chen, Xinyu Huang, Yukang Chen, Feng Yan, Zhaoyang Zeng, Hao Zhang, Feng Li, Jie Yang, Hongyang Li, Qing Jiang, and Lei Zhang. Grounded sam: Assembling open-world models for diverse visual tasks, 2024. 2, 3

[61] Robin Rombach, Andreas Blattmann, Dominik Lorenz, Patrick Esser, and Björn Ommer. High-resolution image synthesis with latent diffusion models. In *CVPR*, 2022. 2

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

[63] Arno Schödl, Richard Szeliski, David H Salesin, and Irfan Essa. Video textures. In *PACMCGIT*, 2000. 2, 3

[64] Ruoxi Shi, Hansheng Chen, Zhuoyang Zhang, Minghua Liu, Chao Xu, Xinyue Wei, Linghao Chen, Chong Zeng, and Hao Su. Zero123++: a single image to consistent multi-view diffusion base model, 2023. 2, 3

[65] Uriel Singer, Adam Polyak, Thomas Hayes, Xi Yin, Jie An, Songyang Zhang, Qiyuan Hu, Harry Yang, Oron Ashual, Oran Gafni, et al. Make-a-video: Text-to-video generation without text-video data. *arXiv preprint arXiv:2209.14792*, 2022. 2

[66] Jascha Sohl-Dickstein, Eric Weiss, Niru Maheswaranathan, and Surya Ganguli. Deep unsupervised learning using nonequilibrium thermodynamics. 2015. 2

[67] Yang Song and Stefano Ermon. Generative modeling by estimating gradients of the data distribution. 2019. 2

[68] Ryusuke Sugimoto, Mingming He, Jing Liao, and Pedro V Sander. Water simulation and rendering from a still photograph. In *SIGGRAPH Asia*, 2022. 2, 3, 5

[69] Roman Suvorov, Elizaveta Logacheva, Anton Mashikhin, Anastasia Remizova, Arsenii Ashukha, Aleksei Silvestrov, Naejin Kong, Harshith Goka, Kiwoong Park, and Victor Lempitsky. Resolution-robust large mask inpainting with fourier convolutions. *arXiv preprint arXiv:2109.07161*, 2021. 2, 4

[70] Martin Szummer and Rosalind W. Picard. Temporal texture modeling. In *ICIP*, 1996. 2, 3

[71] Florin-Alexandru Vasluianu, Tim Seizinger, Zhuyun Zhou, Zongwei Wu, Cailian Chen, Radu Timofte, Wei Dong, Han Zhou, Yuqiong Tian, Jun Chen, et al. Ntire 2024 image shadow removal challenge report. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 6547–6570, 2024. 13

[72] Ruben Villegas, Mohammad Babaeizadeh, Pieter-Jan Kindermans, Hernan Moraldo, Han Zhang, Mohammad Taghi Saffar, Santiago Castro, Julius Kunze, and Dumitru Erhan. Phenaki: Variable length video generation from open domain textual description. *arXiv preprint arXiv:2210.02399*, 2022. 2

[73] Jiawei Wang, Yuchen Zhang, Jiaxin Zou, Yan Zeng, Guoqiang Wei, Liping Yuan, and Hang Li. Boximator: Generating rich and controllable motions for video synthesis. *arXiv preprint arXiv:2402.01566*, 2024. 2

[74] Shuzhe Wang, Vincent Leroy, Yohann Cabon, Boris Chidlovskii, and Jerome Revaud. Dust3r: Geometric 3d vision made easy. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 20697–20709, 2024. 2, 4

[75] Xiang Wang, Hangjie Yuan, Shiwei Zhang, Dayou Chen, Jiuniu Wang, Yingya Zhang, Yujun Shen, Deli Zhao, and Jingren Zhou. Videocomposer: Compositional video synthesis with motion controllability. *arXiv preprint arXiv:2306.02018*, 2023. 3

[76] Yaohui Wang, Xinyuan Chen, Xin Ma, Shangchen Zhou, Ziqi Huang, Yi Wang, Ceyuan Yang, Yinan He, Jiashuo Yu, Peiqing Yang, et al. Lavie: High-quality video generation with cascaded latent diffusion models. *arXiv preprint arXiv:2309.15103*, 2023. 2

[77] Li-Yi Wei and Marc Levoy. Fast texture synthesis using tree-structured vector quantization. In *PACMCGIT*, 2000. 2, 3

[78] Yuxi Wei, Zi Wang, Yifan Lu, Chenxin Xu, Changxing Liu, Hao Zhao, Siheng Chen, and Yanfeng Wang. Editable scene simulation for autonomous driving via llm-agent collaboration. In *The First Workshop on Populating Empty Cities– Virtual Humans for Robotics and Autonomous Driving at CVPR 2024, 2nd Round*. 5

[79] Chung-Yi Weng, Brian Curless, and Ira Kemelmacher-Shlizerman. Photo wake-up: 3d character animation from a single photo. In *CVPR*, 2019. 3

[80] Weijia Wu, Zhuang Li, Yuchao Gu, Rui Zhao, Yefei He, David Junhao Zhang, Mike Zheng Shou, Yan Li, Tingting Gao, and Di Zhang. Draganything: Motion control for anything using entity representation. In *European Conference on Computer Vision*, pages 331–348. Springer, 2024. 5

[81] Hongchi Xia, Zhi-Hao Lin, Wei-Chiu Ma, and Shenlong Wang. Video2game: Real-time, interactive, realistic and browser-compatible environment from a single video. In *CVPR*, 2024. 5

[82] Hongchi Xia, Zhi-Hao Lin, Wei-Chiu Ma, and Shenlong Wang. Video2game: Real-time interactive realistic and browser-compatible environment from a single video. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 4578–4588, 2024. 4

[83] Tianyi Xie, Zeshun Zong, Yuxing Qiu, Xuan Li, Yutao Feng, Yin Yang, and Chenfanfu Jiang. Physgaussian: Physics-integrated 3d gaussians for generative dynamics. In *CVPR*, 2024. 5

[84] Tianyi Xie, Zeshun Zong, Yuxing Qiu, Xuan Li, Yutao Feng, Yin Yang, and Chenfanfu Jiang. Physgaussian: Physics-

integrated 3d gaussians for generative dynamics. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 4389–4398, 2024. 2

[85] Jinbo Xing, Menghan Xia, Yong Zhang, Haoxin Chen, Wangbo Yu, Hanyuan Liu, Xintao Wang, Tien-Tsin Wong, and Ying Shan. Dynamicrafter: Animating open-domain images with video diffusion priors. *arXiv preprint arXiv:2310.12190*, 2023. 2

[86] Jiale Xu, Weihao Cheng, Yiming Gao, Xintao Wang, Shenghua Gao, and Ying Shan. Instantmesh: Efficient 3d mesh generation from a single image with sparse-view large reconstruction models. *arXiv preprint arXiv:2404.07191*, 2024. 2, 3

[87] Tianfan Xue, Jiajun Wu, Katherine L Bouman, and William T Freeman. Visual dynamics: Stochastic future generation via layered cross convolutional networks. *T-PAMI*, 2018. 2, 3

[88] Chenggang Yan, Biyao Shao, Hao Zhao, Ruixin Ning, Yongdong Zhang, and Feng Xu. 3d room layout estimation from a single rgb image. *IEEE Transactions on Multimedia*, 22 (11):3014–3024, 2020. 2

[89] Hanshu Yan, Jun Hao Liew, Long Mai, Shanchuan Lin, and Jiashi Feng. Magicprop: Diffusion-based video editing via motion-aware appearance propagation. *arXiv preprint arXiv:2309.00908*, 2023. 2

[90] Ruihan Yang, Prakhar Srivastava, and Stephan Mandt. Diffusion probabilistic modeling for video generation. *Entropy*, 25(10):1469, 2023. 2

[91] Ze Yang, Yun Chen, Jingkang Wang, Sivabalan Manivasagam, Wei-Chiu Ma, Anqi Joyce Yang, and Raquel Urtasun. Unisim: A neural closed-loop sensor simulator. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 1389–1399, 2023. 5

[92] Zhengyuan Yang, Linjie Li, Kevin Lin, Jianfeng Wang, Chung-Ching Lin, Zicheng Liu, and Lijuan Wang. The dawn of lmms: Preliminary explorations with gpt-4v (ision). *arXiv preprint arXiv:2309.17421*, 9(1):1, 2023. 3

[93] Lijun Yu, José Lezama, Nitesh B Gundavarapu, Luca Versari, Kihyuk Sohn, David Minnen, Yong Cheng, Agrim Gupta, Xiuye Gu, Alexander G Hauptmann, et al. Language model beats diffusion–tokenizer is key to visual generation. *arXiv preprint arXiv:2310.05737*, 2023. 2

[94] Albert J Zhai, Yuan Shen, Emily Y Chen, Gloria X Wang, Xinlei Wang, Sheng Wang, Kaiyu Guan, and Shenlong Wang. Physical property understanding from language-embedded feature fields. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 28296–28305, 2024. 18

[95] Shiwei Zhang, Jiayu Wang, Yingya Zhang, Kang Zhao, Hangjie Yuan, Zhiwu Qing, Xiang Wang, Deli Zhao, and Jingren Zhou. I2vgen-xl: High-quality image-to-video synthesis via cascaded diffusion models. *arXiv preprint arXiv:2311.04145*, 2023. 2

[96] 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 *European Conference on Computer Vision*, pages 388–406. Springer, 2025. 2, 5

[97] Xianling Zhang, Nathan Tseng, Ameerah Syed, Rohan Bhasin, and Nikita Jaipuria. Simbar: Single image-based scene relighting for effective data augmentation for automated driving vision tasks. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 3718–3728, 2022. 2

[98] Jian Zhao and Hui Zhang. Thin-plate spline motion model for image animation. In *CVPR*, 2022. 3

[99] Yupeng Zheng, Chengliang Zhong, Pengfei Li, Huan-ang Gao, Yuhang Zheng, Bu Jin, Ling Wang, Hao Zhao, Guyue Zhou, Qichao Zhang, et al. Steps: Joint self-supervised nighttime image enhancement and depth estimation. In *2023 IEEE International Conference on Robotics and Automation (ICRA)*, pages 4916–4923. IEEE, 2023. 2

[100] Chengliang Zhong, Yuhang Zheng, Yupeng Zheng, Hao Zhao, Li Yi, Xiaodong Mu, Ling Wang, Pengfei Li, Guyue Zhou, Chao Yang, et al. 3d implicit transporter for temporally consistent keypoint discovery. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 3869–3880, 2023. 2

[101] Leisheng Zhong, Yu Zhang, Hao Zhao, An Chang, Wenhao Xiang, Shunli Zhang, and Li Zhang. Seeing through the occluders: Robust monocular 6-dof object pose tracking via model-guided video object segmentation. *IEEE Robotics and Automation Letters*, 5(4):5159–5166, 2020. 2

PhysGen3D: Crafting a Miniature Interactive World from a Single Image

Supplementary Material

In the supplemental materials, we present additional details about our PhysGen3D framework App. A, more details of our experimental design App. B.1, more quantitative and qualitative results App. B.3, and various applications of our system App. B.4. Furthermore, we invite the reviewers to check a local webpage in the supplemental materials accessed by `index.html`, to see our generated videos.

A. Additional Details of PhysGen3D

We provide additional details about our framework, specifically on how we handle multiple object occlusions during the mesh generation stage, how we address background completion concerning objects and their shadows, the detailed prompt used in physics reasoning, and further specifics about the physical simulator utilized in our approach.

A.1. Mesh Generation

To reconstruct a 3D foreground object, we require a complete and clearly segmented object image o^i. For scenarios with multiple object occlusions, we employ an iterative inpainting and segmentation strategy, as illustrated in Fig. 10. We first identify all the target objects using GPT-4o. In cases where occlusions are detected, the objects are segmented and inpainted sequentially, progressing from the foreground to the background. Each subsequent segmentation step builds upon the removal of previously processed objects, ensuring accurate and unobstructed reconstruction.

A.2. Background Handling

Shadow significantly impacts the quality of background inpainting if not masked properly. Existing shadow removal methods [42, 51, 71] typically detect and remove all shadows indiscriminately. However, our goal is to remove only the shadow related to a specific object. To achieve this, we adopt a straightforward method: we first segment regions where brightness values fall below a certain threshold to identify shadows. For each object, we determine the largest connected component that includes both the object and its shadow. Then, we dilate this mask with a kernel of size 50 and apply inpainting. Developing more adaptable, per-object shadow removal techniques is left as future work.

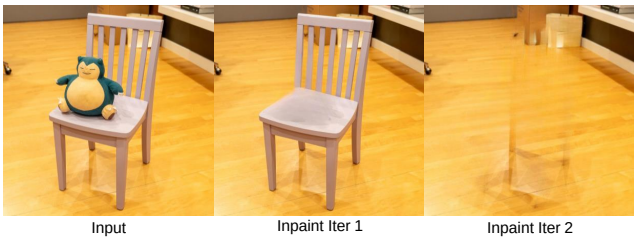

Figure 10. **Iterative Inpainting.** Left: Input image. Middle: Inpainting result after 1 iteration, where the toy is masked and inpainted. Right: Inpainting result after 2 iterations, where the chair is masked and inpainted. The second result is used as background.

A.3. Physics Reasoning

basicstyle=, `backgroundcolor=` `frame=single, breaklines=true, breakatwhitespace=true, columns=flexible,`

We use GPT-4o to reason the physical parameters for each object and the surface. The prompt and an example answer are as follows. [caption=Prompt used for GPT-4o physics reasoning] Answer each question for each object in the picture, using one word or number, separated by commas. For numbers, do not use scientific notation. Provide answers in the following format for each object: 'Object number, name, density in kg/m^3, $Young's modulus (soft/medium/hard)$, $size in meters$, $requires$ $Soft: Materials like plush toys, foam, or fabric. Medium:$ $Materials like rubber or soft plastic. Hard$: $Materials like wood, metal, or hard plastic) What is its size in meters? Do$

Estimate the roughness of the supporting surface in the picture, such as tables, floors, or any other horizontal surfaces that can act as supports. Provide answers in roughness value (0 to 1, where 0 = perfectly smooth and 1 = extremely rough)'.

[caption=Example answers from GPT-4o] 1, camera$_model$, 200, $soft$, 0.15, $yes2$, $camera$, 2700, $hard$, 0.20, $no0.2$

In our observation, GPT-4 often provides unstable results for the exact value of Young's modulus, with discrepancies spanning several orders of magnitude. To address this, we defined three categories—**soft**, **medium**, and **hard**—to guide GPT's classification. In the simulator, the elasticity E does not directly correspond to the real Young's modulus. Based on experience, we associate the three categories with $E = 5 \times 10^4$, $E = 5 \times 10^5$, and $E = 5 \times 10^6$, respectively.

A.4. Dynamics Simulation

For simulation stability, we fix the size of the simulator to 2 and the resolution to 256. Since the target object's scale varies from several centimeters to tens of meters, we align the object with the reconstructed scene and fit it into the simulator. To simulate real physics, we scale the physical parameters accordingly. Suppose the reasoned real size of the object is s_0, and the scaled mesh has size s'. Then, the scaling factor is $k = \frac{s'}{s_0}$. In the simulator, we set gravity to $g' = k \times g_0 = k \times 9.8$. The elasticity of each object is also scaled: $E'_i = \frac{E_i}{k}$. (According to dimensional analysis,

Young's modulus is inversely proportional to the scale of length.)

We use Taichi Elements [31–33] for Material Point Method (MPM) simulations and modify it to support inhomogeneous materials. MPM is a computational technique used to simulate the behavior of continuum materials. The governing equation of motion is:

$$\rho \frac{D\mathbf{v}}{Dt} = \nabla \cdot \boldsymbol{\sigma} + \mathbf{f}_{\text{ext}},$$

where:
- ρ: Density of the material,
- $\mathbf{v}$: Velocity field,
- $\boldsymbol{\sigma}$: Cauchy stress tensor,
- $\mathbf{f}_{\text{ext}}$: External forces per unit volume.

To be specific, MPM combines the strengths of Lagrangian and Eulerian methods by representing materials as discrete particles while performing computations on a background grid. The key steps of MPM are particle-to-grid (p2g) and grid-to-particle (g2p) transfers.

Particle-to-Grid (p2g) Transfer. This step transfers particle properties (mass, momentum, etc.) to the grid.

Mass Transfer. Grid mass is computed by distributing particle mass m_p to nearby grid nodes using weighting functions w:

$$m_i = \sum_p w(x_p - x_i) m_p,$$

where:
- $m_p = \rho_p V_p$: Particle mass (density ρ_p, volume V_p),
- w: Quadratic kernel for interpolation.

Momentum Transfer. Momentum is transferred to the grid using the same weight:

$$\mathbf{v}_i = \frac{\sum_p w(x_p - x_i)\mathbf{v}_p m_p}{m_i},$$

where:
- $\mathbf{v}_i$: Grid velocity,
- $\mathbf{v}_p$: Particle velocity.

Stress Contribution. The stress tensor $\boldsymbol{\sigma}$ contributes force to the grid momentum. Using the deformation gradient F, the stress is defined as:

$$\boldsymbol{\sigma} = 2\mu(F - \mathbf{R})F^\top + \lambda J(J-1)\mathbf{I},$$

where:
- μ and λ: Lamé parameters,
- F: Deformation gradient,
- $\mathbf{R}$: Rotation matrix from SVD ($F = \mathbf{RS}$),
- $J = \det(F)$: Determinant of F,
- $\mathbf{I}$: Identity matrix.

The Lamé parameters λ and μ are computed from Young's modulus E and Poisson's ratio ν as follows:

$$\lambda = \frac{E\nu}{(1+\nu)(1-2\nu)}$$

$$\mu = \frac{E}{2(1+\nu)}$$

where:
- E: Young's modulus, which describes the material's stiffness,
- ν: Poisson's ratio, which defines the ratio of lateral strain to axial strain.

Grid Velocity Update. The grid force due to stress is given by:

$$\mathbf{f}_i = -\sum_p w'(x_p - x_i) V_p \boldsymbol{\sigma}_p.$$

Newton's second law updates grid velocities:

$$\mathbf{v}_i^{n+1} = \mathbf{v}_i^n + \Delta t \frac{\mathbf{f}_i}{m_i},$$

where Δt is the time step.

Grid-to-Particle (g2p) Transfer This step interpolates updated grid data back to particles and updates their states (e.g., velocity, deformation).

Velocity Interpolation. Particle velocities are updated by interpolating grid velocities:

$$\mathbf{v}_p^{n+1} = \mathbf{v}_p^n + \sum_i w(x_p - x_i)\mathbf{v}_i^{n+1}.$$

Affine Velocity Field. Affine velocity updates capture velocity gradients from the grid:

$$\mathbf{C}_p = \sum_i 4\frac{w(x_p - x_i)}{\Delta x}\mathbf{v}_i \otimes (\mathbf{x}_i - \mathbf{x}_p).$$

Deformation Gradient Update. The deformation gradient F_p evolves based on the velocity gradient:

$$F_p^{n+1} = (\mathbf{I} + \Delta t \mathbf{C}_p)F_p^n,$$

where $\mathbf{I}$ is the identity matrix.

Advection. Finally, particles are advected using updated velocities:

$$\mathbf{x}_p^{n+1} = \mathbf{x}_p^n + \Delta t \mathbf{v}_p^{n+1}.$$

B. Additional Details of Experiments

Our experiments are designed to compare with the most competitive baselines using multiple evaluation metrics, including human evaluation and GPT-based evaluation. Due to page limitations in the main paper, we provide detailed information about the experimental settings, evaluation metrics, and additional results here.

B.1. Experiments Settings

In the comparative experiment between our method and baseline generative models, we tried our best to ensure they shared the same generation goal. For our method, we manually assigned an initial 3D velocity to each object. To "interpret" this into text, we described the corresponding dynamics and converted them into prompts such as, *"The elephant hops up and falls onto the ground"* or *"The book falls and the orange rolls forward."* All three baseline models were prompted with the same text. Additionally, Kling supports "motion brush" inputs, which were provided alongside the textual prompt. Fig. 11 illustrates examples of "motion brush" inputs, where we manually set the stable parts, movable parts, and their trajectory.

B.2. Evaluation

In our main paper, we only present the quantitative results of human evaluation. Here, we conduct further experiments using GPT-4 and provide the details.

Human Evaluation. We designed a questionnaire to conduct human evaluation, as illustrated in Fig. 12. A total of 31 participants were recruited to complete the 27-page questionnaire. At the beginning, we provided an explanation of video generation models to ensure that participants had a clear understanding of the task. Each page of the questionnaire contains an initial reference image, accompanied by a text prompt describing the expected behavior in the video (e.g., "Red apple rolls on the table"). Four videos are presented on each page in a random order, all corresponding to the same initial condition and text prompt. Participants are instructed to assess each video based on three dimensions. This design ensures a fair, consistent, and comprehensive evaluation process.

GPT-4o Evaluation. To assess the quality of the generated videos, we also conducted evaluations using GPT-4o for both our results and the baselines. The prompt is as follows: [caption=Prompt used for GPT-4o evaluation] I would like you to evaluate the quality of four generated videos based on the following criteria: physical realism, photorealism, and semantic consistency. The evaluation will be based on 10 evenly sampled frames from each video. Given the original image and the following instructions: 'instructions', please evaluate the quality of each video on the three criteria mentioned above. Note that: Physical Realism measures how realistically the video follows the physical rules and whether the video represents real physical properties like elasticity and friction. To discourage completely stable video generation, we instruct respondents to penalize such cases. Photorealism assesses the overall visual quality of the video, including the presence of visual artifacts, discontinuities, and how accurately the video replicates details of light, shadow, texture, and materials. Semantic Consistency evaluates how well the content of the generated video aligns with the intended meaning of the text prompt. Please provide the following details for each video, scores should be ranging from 0-1, with 1 to be the best: Video 1: Physical Realism Score: [a score]; Photorealism Score: [a score]; Semantic Consistency Score: [a score] Video 2: Physical Realism Score: [a score]; Photorealism Score: [a score]; Semantic Consistency Score: [a score] Video 3: Physical Realism Score: [a score]; Photorealism Score: [a score]; Semantic Consistency Score: [a score] Video 4: Physical Realism Score: [a score]; Photorealism Score: [a score]; Semantic Consistency Score: [a score] Note that your output should strictly follows the above format, with a ';' after each score. Do not give any other explanations. The first image is the input image. input image Here are 10 evenly spaced frames from the generated video number idx + 1. generated frames

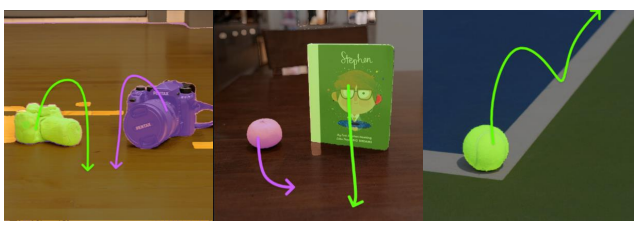

Figure 11. **Motion brush input for Kling.** In all cases, we manually define the motion for each object by identifying the movable part and drawing its trajectory. Additionally, we specify the stable part of the object.

B.3. Additional Results

show that both methods introduce unrealistic deformations. DragAnything sometimes fails to maintain a stable background, even when manually set. MOFA demonstrates better motion control but lacks realism as well. See the table below for quantitative results. We provide additional quantitative and qualitative results of our experiments.

Human Evaluation Results. We analyze the human evaluation scores further in Fig. 13. The distribution of scores indicates that participants generally agree that most of our results are both physically realistic and semantically consistent. Our method significantly outperforms baseline generative models on these two criteria. However, the four models perform comparably in terms of photorealism.

Additional Qualitative Results. Here, we present additional qualitative results in Fig. 14. The first row demonstrates the "sandy" effect, where the material of the teddy bear is transformed into sand. The last row illustrates a multi-object collision scenario, where three apples collide with one another. More results are available in video format on our supplementary webpage.

Fig. 15 shows the results after VEnhancer's post-processing. Although VEnhancer recovers fine

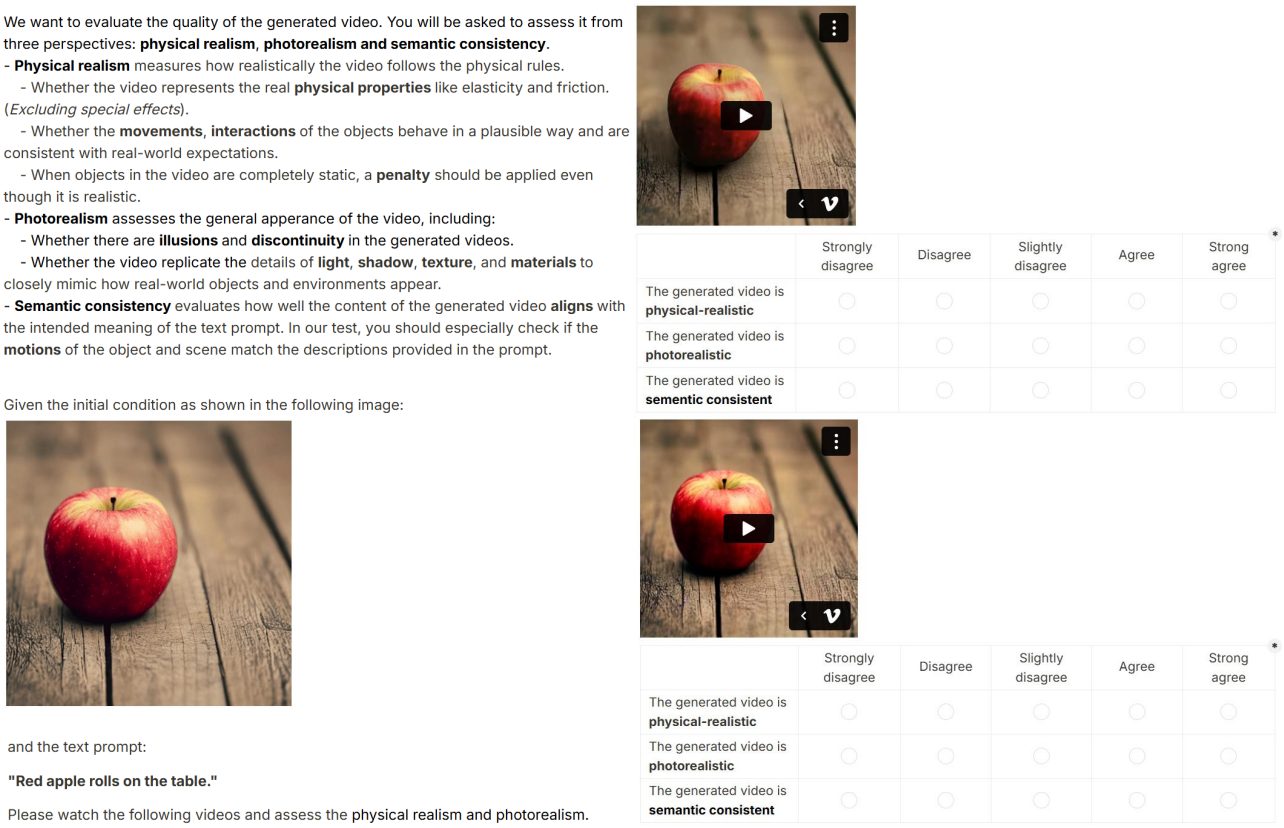

Figure 12. **An example page of human evaluation questionnaire.** In each page of the questionnaire, we explain the criteria in detail. We provide the input image, the text prompt and four generated videos in a random order. Each video is followed by a evaluation matrix on a five-point scale, from strongly disagree (1) to strongly agree (5).

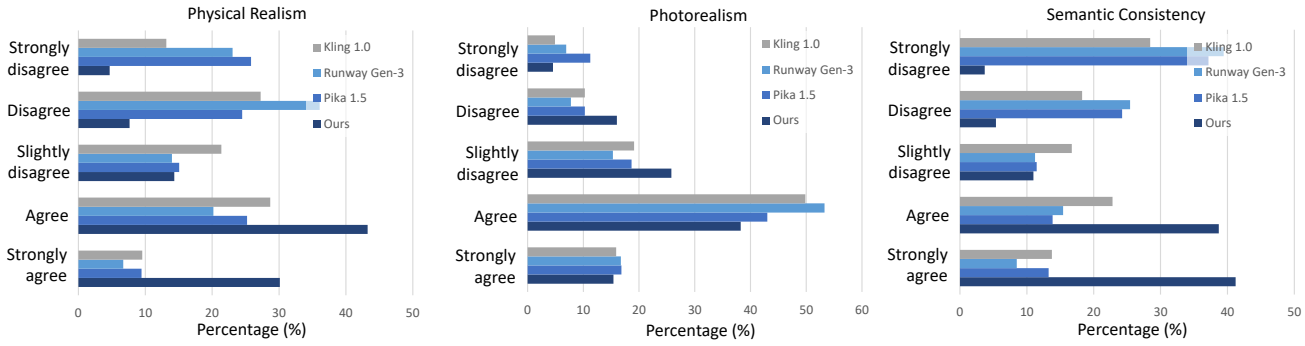

Figure 13. **Human evaluation score distribution.** Score distribution shows our method's superiority in physical realism and semantic consistency, with comparable performance across models in photorealism.

details, it can also introduce hallucinations. This illustrates a fundamental trade-off between photorealism and physical accuracy: integrating diffusion models into the pipeline leverages their strong priors to compensate for reconstruction and rendering errors, but it cannot guarantee adherence to real-world physics.

Fig. 16 shows the results of two open-sourced diffusion models, MOFA-Video and DragAnything. Both methods introduce unrealistic deformations: DragAnything sometimes fails to maintain a stable background, even when manually set. MOFA demonstrates better motion control but lacks realism as well. Quantitative results of VBench scores in the main paper support these findings.

Figure 14. **More qualitative results.** The first row demonstrates the **"sandy" effect**, transforming the teddy bear's material into sand, while the second and third rows showcase **bouncing** and **rolling** effects, respectively. The fourth row illustrates a **multi-object collision** scenario, with three apples colliding with one another, and the final row highlights the system's ability to **generate a video from a painting**.

Figure 15. **Qualitative comparison of VEnhancer.** After post-processing by VEnhancer, more details are recovered and the video appears to be more photorealistic.

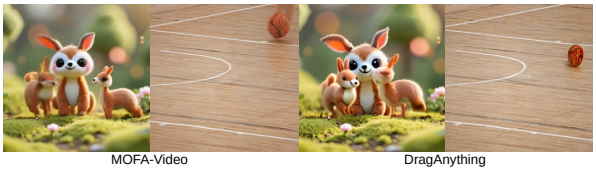

Figure 16. **Qualitative results of MOFA-Video and DragAnything.** These two open-sourced diffusion models fail to keep background consistent and produce unrealistic deformations.

B.4. Applications

Our video generation framework, PhysGen3D, enables a range of exciting applications through its explicit representation. Here are just a few of the compelling use cases our system supports:

Camera controls. PhysGen3D's 3D scene representation inherently supports novel view synthesis. We demonstrate this capability (see figure below) by extending our method with minimal modifications: (1) outpainting and meshing the background and (2) rendering from novel views. Results in Fig. 18 show good consistency across views while

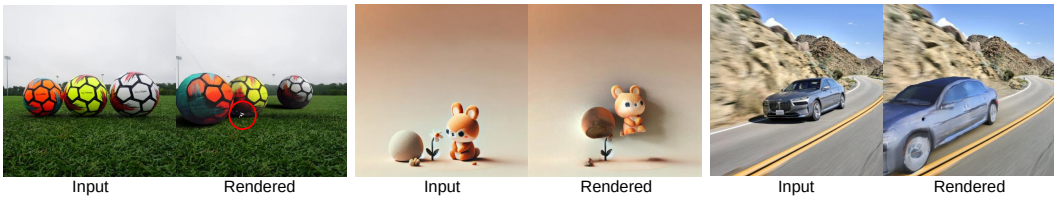

Figure 17. **More On Limitations.** The left two images show simulation failures, where unwanted floating points appear in the final rendering results. The middle two images show reconstruction failures, where the wall is recognized as ground by mistake. The right two images depict texture optimization failures, where the car fails to accurately reproduce the real roughness and metallic properties, resulting in an unrealistic appearance.

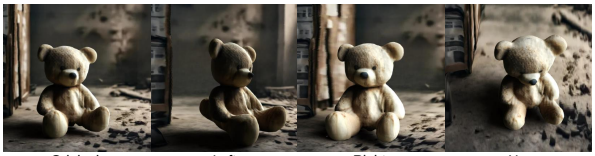

Figure 18. **Camera controls.** We provide a case demonstrating the potential to perform camera controls on above our pipeline. The left one is the only input image. The right three images are generated with outpaiting and reconstruction.

maintaining environmental coherence.

Generate Video from Paintings. Thanks to the generalization ability of our interactive 3D world reconstruction pipeline, our method can extend beyond real photos to accommodate other types of inputs, such as generated images and paintings. The final row of Fig. 14 demonstrates the generation of a video from a painting.

C. Limitations

In the main text, we present three failure cases, each highlighting a specific type of error in perception, simulation, and rendering. Fig. 17 illustrates additional failures. One involves incorrectly reconstructed meshes with unwanted floating points. Although we have implemented floating point removal during rendering, some points are too close to the object to be detected. Another failure involves material that is incorrectly estimated. The reflectance behavior of cars poses a challenging optimization target, and inaccuracies in inverse rendering result in unrealistic renderings. Failures or inaccuracies may also occur in depth and light estimation. However, these modules are relatively mature, and such errors are comparatively rare.

Many of these failures stem from the inherently ill-posed nature of the task, as reconstructing the full geometry, physics, and textures from partial scene observations requires substantial prior knowledge.

Currently, we only support a single collider surface, such as the ground or a table. However, our pipeline has the potential to set all stable components as colliders. Additionally, each object is currently homogeneous in density and elasticity. In the future, we may assign different materials to different parts of an object, as demonstrated in [94].

Overall, our method is designed for object-centric scenes, excelling at mimicking real-world physics for rigid and deformable objects. It also supports a variety of edits and effects. However, reconstructing entire scenes for more complex scenarios remains an open challenge.