WonderPlay: Dynamic 3D Scene Generation from a Single Image and Actions

Supplementary Material

A. Additional Evaluation

Additional ablation. We provide additional ablation on diffusion parameters including s_1, s_2, and γ in Table S1. The performances of different configurations slightly drop compared to our optimal values. In particular, lower s_1 and s_2 insufficiently leverage video priors, while higher values could weaken adherence to the physics simulation. Similarly, lower γ under-leverage video priors.

Staged evaluation. We design a staged evaluation with increasing complexity in a series of scenes (Figure S1): a simple rigid ball falling onto a desk (stage 1), make the ball elastoplastic (stage 2), replace the desk with water to form multiphysics (stage 3), and replace the ball with a duck for more complex shape (stage 4). As shown in the diagrams in Figure S2, while baseline methods perform well in early stages with simple physics, our method significantly outperforms baselines in later stages where the scenes involve complex physics and geometry.

B. Technical Details

B.1. Additional Implementation Details

When conditioning the video generator with simulated dynamics, we use the standard resolution and time values: $H = 480$, $W = 720$, with $T = 48$ frames (in total 49 frames output). For sampling, we use a DDIM [57] scheduler and iterate $S = 25$ steps on the warped noise for the final video output. We empirically set degradation factor $\gamma = 0.4$ and apply appearance signal at $s_1 = 21, s_2 = 18$ diffusion steps , as this combination usually provides the optimal results.

B.2. Reconstructing Background

In a nutshell, the initial scene background $\mathcal{B}_0$ is generated by decomposing the input image $\mathbf{I}$ into several image layers, unprojecting all pixels in each layer to 3D space with estimated depth [31], followed by a photometric optimization to match the rendering with the input image $\mathbf{I}$ via differentiable rendering [32]. We refer the reader to Yu et al. [72] for more details of the generation process.

B.3. Reconstructing Topological Gaussian Surfels

To reconstruct the 3D objects by the topological Gaussian surfels from the input image $\mathbf{I}$, we first segment the object image by the Segment Anything Model [33] and then we apply an image-to-mesh generation model InstantMesh [69]. In addition to the mesh, InstantMesh also generates multi-view object images $\{\mathbf{I}_i\}$ at fixed viewpoints as intermediate outputs. We bind a Gaussian surfel to each of the mesh vertices. Specifically, we first initializing a Gaussian surfel at a vertex with the vertex normal and the vertex color, and then we optimize the Gaussian surfel parameters so that the rendered images matches the multi-view images $\{\mathbf{I}_i\}$ via differentiable rendering [32, 72].

However, up to here the topological Gaussian surfels are still in a canonical coordinate frame. We need to register each of objects back to the scene coordinate frame. To do this, we first estimate the object orientation by DUSt3R [61], and then we solve for a scale s and a 3D translation $\mathbf{T}$ by least square to align the two coordinate frames. This requires us to find 3D correspondences to form the least square objective. We sample 3D points in the scene coordinate frame by first sampling pixels in $\mathbf{I}$ within the object segment, and then unprojecting the object pixels to 3D with the estimated depth [31], similar to the background. To sample 3D points in the object canonical frame, we sample object pixels from the image rendered from the object representation with the DUSt3R-estimated pose. Each of these pixels uniquely correspond to a 3D point in the object canonical frame.

For stabler simulation, we also adopt the internal filling technique as in PhysGaussian [67].

B.4. Material and Physics Solvers

We consider homogeneous uniform materials, i.e., $\mathbf{m}$ is constant within an object. We follow Liu et al. [43] to estimate the values of the material parameters $\mathbf{m}$ by a Vision-Language Model (VLM) with optional manual adjustment for physical plausibility during simulation.

Here we provide further complementary information on each object material model and their solvers. We consider homogeneous uniform materials, i.e., $\mathbf{m}$ is constant within an object. To model an object, we follow Liu et al. [43] to do a 6-way classification (rigid, elastic, cloth, smoke, liquid, and granular) by a VLM, and estimate the values of the material parameters $\mathbf{m}$ by the VLM with optional manual adjustment for physical plausibility during simulation. The material models and solvers are as follows.

Rigid body. We model a rigid object as a strictly undeformable mesh without internal links. The material properties $\mathbf{m}$ of a rigid object includes the density ρ and the friction coefficient k. Recall that our topological Gaussian surfels are given by: $\mathcal{O}_t = \{\mathbf{E}, \mathbf{v}_t, \mathbf{p}^{\text{O}}_t, \mathbf{q}^{\text{O}}_t, \mathbf{s}^{\text{O}}_t, \mathbf{o}^{\text{O}}_t, \mathbf{c}^{\text{O}}_t\}$, where the edge matrix $\mathbf{E} \in \{0,1\}^{N_{\text{O}} \times N_{\text{O}}}$ indicates the topological connectivity of the surfels, and $\mathbf{v}_t \in \mathbb{R}^{3N_{\text{O}}}$ denotes the velocity. They can be seen as a super-set of a mesh that has $\mathbf{E}$ and $\mathbf{p}$. Therefore, we can directly apply a rigid body

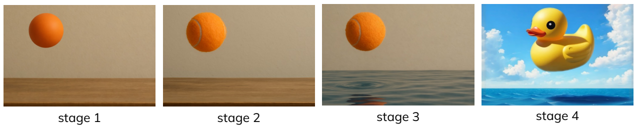

Figure S1. Scenes with increasing complexity. The first scene involves a rigid ball falling onto a rigid plane. The second replaces the rigid ball with a soft ball. The third scene replaces the rigid plane with water surface to include multi-physics. The fourth scene include an object with complex geometry.

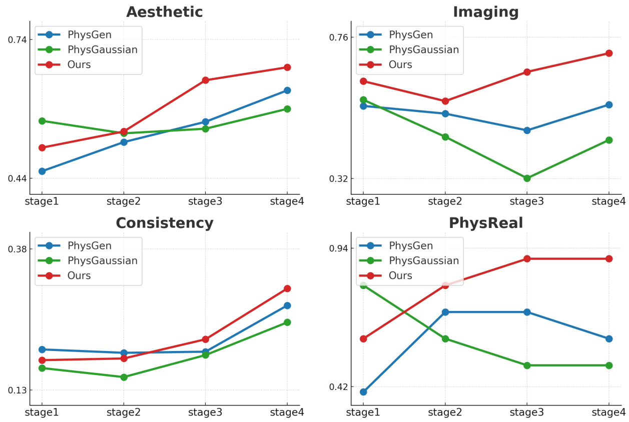

Figure S2. Quantitative results on the four stages (scenes with increasing complexity).

Methods	Imaging (↑)	Aesthetic (↑)	Motion (↑)	Consistency (↑)	PhysReal (↑)
Ours	**0.695**	0.610	**0.995**	0.217	**0.700**
Ours w/o RGB	0.673	0.601	0.993	0.212	0.670
Ours w/o flow	0.574	0.587	0.994	0.213	0.650
Coarse simulation	0.552	0.577	**0.995**	0.197	0.500
$s_1{=}19, s_2{=}16$	0.610	0.571	0.994	0.215	0.650
$s_1{=}23, s_2{=}20$	0.683	0.581	**0.995**	0.217	0.690
$\gamma = 0.3$	0.662	0.571	0.994	0.217	0.670

Table S1. Ablation study on conditioning signals and diffusion hyper-parameters.

solver to simulate our rigid objects. We adopt a rigid body solver based on shape matching [47]. At each simulation time step, the rigid solver uses action forces to update the dynamics attributes, and then detects collisions among rigid objects and the background using connectivity information $\mathbf{E}$ to resolve penetrations.

Elastic, liquid, and granular materials. We model these materials with continuum mechanics and simulate them using a Material-Point-Method (MPM) solver [30], similar to PhysGaussian [67]. The material properties $\mathbf{m}$ include the density ρ, Young's modulus E, and Poisson's ratio ν. For granular material, the material properties also include the friction angle θ. The physics solver is built upon MPM [30], a hybrid Eulerian-Langrangian method. It simulates based on both particles and a spatial grid. As mentioned above, we densely sample particles inside the object in addition to the surface surfels. In each simulation time step, an MPM solver computes the momentum of each object particle (surfel) to update the dynamics attributes. In detail, the momentum of each particle is transferred to the grid within a particle-to-grid step, to further compute the terms like deformation gradient. These updates are back propagated into particles through a grid-to-particle process to update particle dynamics properties like position and velocity.

Cloth and smoke. We model smoke and cloth with only particles and employ the Position-Based Dynamics (PBD) solver [48] for these effects. The material properties $\mathbf{m}$ for cloth includes density ρ and stretch/bending compliance p. The material properties for smoke includes ρ and viscosity coefficient μ. We also densely sample particles inside smoke. Unlike MPM method, PBD method directly models the positions of each particle through a list of inequality and non-equality constraints. In each time step, PBD solver solves each constraint sequentially and directly update the particle's position which is then used to update the velocity. The constraints for smoke include incompressibility [44]; the constraints for cloth include stretch and bending compliance. We refer the read to Bender et al. [9] for more information.

B.5. Simulation Parameters

Different solvers rely on the different sets of physical parameters. Here we provide a table of all the parameters we set in the physical simulation process in Tab S2, also with their default values. In simulation these parameters can be roughly estimated with a VLM and optional manual adjustment, as long as the simulation results are reasonable.

B.6. Rendering Simulated Dynamics

Upon physical simulation, we need to further map the simulated outputs to the Gaussian surfels for rendering the coarse dynamics. For rigid body objects, since each Gaussian surfel is initialized from one mesh vertex and the rigid body solver runs on the mesh representation, we can directly update each surfel's position with simulation results. For particle-based MPM and PBD solvers, during the initial sampling process, we record the mapping of each Gaussian surfel to the nearest 10 sampled particles. At each simulation step, we use the average of position updates of these nearest particles to update the position of the corresponding Gaussian surfel.

Parameter	Default Value
General simulation	
Step time	$1e^{-2}$
Sub-steps number	10
Sampled particle size	$1e^{-2}$
Gravity	$(0, 0, -9.8)$
Rigid body solver	
friction coefficient	0.1
MPM solver	
Grid density	128
Elastic material Young's modulus	$3e^{5}$
Elastic material Poisson's ratio	0.2
Liquid material Young's modulus	$1e^{7}$
Liquid material Poisson's ratio	0.2
Granular material Young's modulus	$1e^{6}$
Granular material Poisson's ratio	0.2
Granular material Friction angle	45
PBD solver	
Cloth material stretch compliance	$1e^{-7}$
Cloth material bending compliance	$1e^{-5}$
Smoke material viscosity coefficient	0.1

Table S2. Simulation parameters and default values