

Supplementary Material for: PhysGM: Large Physical Gaussian Model for Feed-Forward 4D Synthesis

Appendix

A. More Details on Implementation

A.1. Compare with Other Methods

Table 1 provides a qualitative comparison between our method, PhysGM, and other state-of-the-art approaches [5, 9, 10, 12, 17, 20, 21]. We evaluate each method across five critical dimensions: two concerning input requirements (the need for pre-optimized 3D Gaussians or pre-defined physical parameter) and three concerning core capabilities (generalizability, independence from Large Language Models, and inference speed). The comparison highlights that our approach is the only one to operate without these stringent prerequisites. PhysGM simultaneously achieves strong generalization and maintains a very short inference time of under 30 seconds.

A.2. Simulation Details

This section elaborates on the key parameters used to configure our Material Point Method (MPM) simulations, as referenced in the main text. The configuration is detailed below, categorized by function.

MPM Grid Resolution The simulation domain is discretized into a background grid of $200 * 200 * 200$ cells. This grid is fundamental to the MPM algorithm for computing particle interactions and mapping data between particles and the grid.

Camera Position For different objects, the camera is initialized at an azimuth of -45 or 135 degrees, an elevation of 0 degrees, and a radius of 1.8 or 1.3 units.

Camera Motion The camera is configured to be static during the simulation.

Other Parameters Gravity is applied in the falling scene, and force in the corresponding direction is applied in the collision scene.

A.3. Training and Evaluation Details

Network Architecture. We employ DINOv3 [13] (ViT-L/16) pre-trained on LVD-1689M as our image encoder,

producing 1024-dimensional features. The transformer backbone consists of 24 layers with a hidden dimension of 1024 and attention head dimension of 64. We use a patch size of 16 and incorporate 3 learnable global tokens for physics prediction. For 3D Gaussian representation, we set the spherical harmonics degree to 0, with near and far planes at 0.001 and 2.0, respectively.

Training Configuration. We train our model on 32 NVIDIA A800 GPUs using a two-stage process for about 3 days in total with a batch size of 8 per GPU. The base learning rate is set to $2 * 10^{-4}$ with AdamW optimizer ($\beta_1 = 0.9$, $\beta_2 = 0.95$, weight decay = 0.05). We employ a cosine learning rate schedule with 5K warmup steps and clip gradients to a maximum norm of 10.0. Mixed precision training is enabled using bfloat16 to accelerate computation.

Data Configuration. During training, input images are resized to 512×512 resolution with square cropping. Each training sample consists of 4 input views and 8 target views, where target views include the input views for consistency. We use 8 workers for data loading with a prefetch factor of 128 to ensure efficient GPU utilization.

Evaluation Protocol. We evaluate on the complete GSO dataset [4] containing 1,009 objects. For each object, we render 32 views with 4 elevation angles (0, 20, 40, 60) and 8 azimuthal angles. During testing, we sample fixed 4 views as input and evaluate reconstruction quality on 8 randomly selected novel views. We report PSNR, SSIM [15], and LPIPS [19] averaged across all test views and objects.

A.4. User Preference Evaluation

To complement quantitative metrics with human perception assessment, we conduct a user study to evaluate the perceptual quality and physical plausibility of generated 4D sequences across different methods.

Study Design. We employ a Four-Alternative Forced Choice (4AFC) protocol, where participants are presented with four videos simultaneously showing the same object simulated by different methods: our PhysGM, and three baseline methods (PhysGaussian [17], OmniPhysGS [9], and DreamGaussian4D [12]). The videos are displayed in

Table 1. Comparison with state-of-the-art methods, highlighting PhysGM’s unique advantages. Unlike prior work, our method eliminates the need for both pre-optimized 3D Gaussian and pre-defined physical parameters. This allows it to achieve strong generalization while maintaining a significantly shorter inference time (< 30 s). “only E” represents that only Young’s modulus is automatically predicted, “only material” represents that only material is automatically predicted.

Method	No Pre-opt. 3D Gaussians	Auto Param Computation	Generalizable	Without LLM	Inference Time
PhysGaussian [17]	×	×	×	✓	-
DreamPhysics [5]	×	only E	×	✓	>0.5 h
PhysDreamer [20]	×	only E	×	✓	>1 h
OmniPhysGS [9]	×	only material	×	✓	>12 h
DreamGaussian4D [12]	✓	×	✓	×	6.5min
Feature Splatting [10]	×	×	×	✓	>1 h
PhysSplat [21]	×	✓	✓	×	<2 min
PhysGM (Ours)	✓	✓	✓	✓	<30s

randomized positions to eliminate order bias. Participants are instructed to select the single video that exhibits the most realistic physical behavior and visual quality, considering factors such as motion naturalness, material response, temporal coherence, and rendering fidelity.

Stimuli and Sampling. We carefully select 5 representative test scenes spanning diverse object categories and physical scenarios (dropping and stretching). For each scene, we generate 4D sequences using all methods with identical input views and physical interaction setups to ensure fair comparison.

Participants and Procedure. We recruited 103 participants comprising graduate students and researchers with backgrounds in computer graphics, computer vision, or related fields. Each participant completed a questionnaire containing 5 comparison trials (one per test scene). Before the formal study, participants underwent a training phase with two practice trials to familiarize themselves with the task and interface. Participants could replay videos multiple times before making their selection and were allowed to take breaks between trials. The entire study took approximately 10 minutes per participant.

Data Validation and Filtering. To ensure data quality, we implemented several validation mechanisms:

- **Attention checks:** Two control trials with obvious quality differences were inserted to identify inattentive participants.
- **Completion time:** Responses completed too quickly (<5 seconds per trial) were flagged.
- **Response consistency:** Participants showing random selection patterns were identified via entropy analysis.

After applying these criteria, we excluded 3 invalid responses (2.9% exclusion rate) due to failed attention checks or suspiciously short completion times, resulting in 100 valid responses for analysis.

User Preference Rate (UPR). We define the User Preference Rate as the percentage of participants who selected a given method as the most realistic:

$$\text{UPR}_m = \frac{1}{S \cdot N} \sum_{s=1}^S \sum_{i=1}^N \mathbf{1}[\text{choice}_{s,i} = m] \times 100\% \quad (1)$$

where m denotes the method, S is the number of test scenes, $N = 100$ is the number of valid participants, and $\mathbf{1}[\cdot]$ is the indicator function. A higher UPR indicates stronger human preference. Under random chance, each method would receive 25% preference rate in a 4AFC setup.

B. Material Constitutive Models

In continuum mechanics and physics-based simulation, a constitutive model (or constitutive equation) is a fundamental mathematical relationship that describes how a material responds to external stimuli. Specifically, it defines the relationship between the internal forces (stress) and the material’s deformation (strain). The choice of a constitutive model is critical as it dictates the material’s behavior—whether it behaves as a rigid solid, an elastic solid, a fluid, or a hyperelastic material like rubber. Our simulation framework employs different constitutive models based on the predicted material class. This allows us to capture a diverse range of dynamic behaviors.

B.1. The Neo-Hookean Model

For materials predicted to be “jelly” or other soft, rubber-like substances, we employ the Neo-Hookean model. This is a classic hyperelastic model, meaning its stress-response is derived from a strain energy density function. It is ideal for capturing large, nonlinear deformations while remaining computationally efficient, making it a staple in computer graphics and simulation. The model’s formulation is based on the statistical mechanics of polymer chains, which accurately describes the behavior of materials like rubber that can stretch significantly without permanent deformation.

The core idea is to split the material’s response into two parts: a part that resists changes in shape (deviatoric) and a part that resists changes in volume (volumetric). This allows for a robust simulation of compressible, soft-bodied dynamics. The model defines the Kirchhoff stress (τ), which is a measure of internal force suitable for large-deformation analysis. The Kirchhoff stress τ for a compressible Neo-Hookean material is given by:

$$\tau = \mu * J^{-2/3} * \text{dev}(\mathbf{B}) + (\lambda/2) * (J^2 - 1) * I, \quad (2)$$

where τ is the Kirchhoff stress tensor. $\mathbf{B} = \mathbf{F}\mathbf{F}^T$ is the left Cauchy-Green deformation tensor, where \mathbf{F} is the deformation gradient and $\text{dev}(\mathbf{B})$ is the deviatoric (volume-preserving) part of \mathbf{B} . $J = \det(\mathbf{F})$ is the determinant of the deformation gradient, representing the volume change. μ and λ are the Lamé parameters, which characterize the material’s stiffness. They are derived from the Young’s modulus (E) and Poisson’s ratio (ν) predicted by our model.

B.2. The Fixed Corotational Constitutive Model

For materials predicted to be “metal” or other similarly stiff elastic solids, we employ the Fixed Corotational (FCR) constitutive model. This model is particularly well-suited for scenarios where a material undergoes large rigid-body motions (i.e., translation and rotation) but experiences only small elastic deformations. The core principle of any corotational model is to decouple the object’s overall rotation from its internal strain. The FCR model begins with the polar decomposition of the deformation gradient $\mathbf{F} = \mathbf{R}\mathbf{S}$, where \mathbf{R} is a pure rotation matrix, and \mathbf{S} is the right stretch tensor, which is symmetric and positive definite. The model defines a linear relationship between the First Piola-Kirchhoff stress (\mathbf{P}) and a measure of strain. The First Piola-Kirchhoff stress is energetically conjugate to the deformation gradient \mathbf{F} and is given by:

$$\mathbf{P} = 2\mu(\mathbf{F} - \mathbf{R}) + \lambda(J - 1)J(\mathbf{F}^{-T}), \quad (3)$$

where \mathbf{P} is the First Piola-Kirchhoff stress tensor. For force calculations within our MPM simulation, we use the Kirchhoff stress (τ). The relationship between Kirchhoff stress and the First Piola-Kirchhoff stress is: $\tau = \mathbf{P}\mathbf{F}^T$.

B.3. The Drucker-Prager Plasticity Model

For materials exhibiting both frictional and cohesive properties, such as sand, snow, and plasticine, we employ the Drucker-Prager elastoplasticity model. This model is ideal for materials whose strength is dependent on the hydrostatic pressure they are under (e.g., sand becomes stronger when compressed). It defines a yield criterion, which is a surface in stress space that separates elastic (temporary) deformation from plastic (permanent) deformation. The core of the model is the predictor-corrector algorithm, also known as return mapping: First, the model assumes the material behaves purely elastically during a time step and calculates a “trial stress”. It then checks if this trial stress lies outside the Drucker-Prager yield surface. If the trial stress is outside the surface (i.e., the material has yielded), the stress is mathematically projected back onto the yield surface. This correction step accounts for the plastic flow and ensures the material’s stress state remains physically plausible. The Drucker-Prager yield criterion defines the boundary between elastic and plastic states. The yield function is given by:

$$f(\tau) = \|\text{dev}(\tau)\| + \alpha * \text{tr}(\tau) - k \leq 0, \quad (4)$$

where τ is the Kirchhoff stress tensor. $\text{dev}(\tau)$ is the deviatoric part of the stress, representing shear. $\|\text{dev}(\tau)\|$ is the Frobenius norm of the deviatoric stress, measuring the magnitude of the shear stress. $\text{tr}(\tau)$ is the trace of the stress, proportional to the hydrostatic pressure (positive for tension, negative for compression). α is a dimensionless friction parameter, controlling how much the material’s strength increases with pressure. k is the cohesion of the material, representing its intrinsic shear strength at zero pressure.

The key insight is that different materials like sand, snow, and plasticine can be simulated with the same underlying model by simply adjusting the cohesion (k) and friction (α) parameters. For instance: Sand ($k = 0.0$) has negligible cohesion; its strength comes almost entirely from inter-particle friction. Snow ($k = 1000.0$) represents an intermediate case with some cohesion. Plasticine ($k = 5000.0$) has significant cohesion, allowing it to hold its shape even without compressive pressure.

C. PhysAssets Dataset Statistics

C.1. Dataset Composition

PhysAssets comprises a comprehensive collection of 3D assets with annotated physical properties. The dataset consists of two main components: a training set containing 49,206 objects aggregated from multiple public repositories (Objaverse [3], OmniObject3D [16], ABO [2], and HSSD [8]), and a held-out test set of 1,009 objects from the Google Scanned Objects (GSO) dataset [4], totaling 50,215 annotated 3D objects. The primary objective of this effort was

Table 2. Material distribution in PhysAssets dataset. The 14 primary materials account for 97% of the dataset, while 32 rare materials provide additional diversity.

Rank	Material	Count	Percentage
1	Plastic	13,696	27.3%
2	Wood	8,443	16.8%
3	Metal	7,353	14.6%
4	Fabric	7,255	14.5%
5	Ceramic	3,023	6.0%
6	Stone	2,135	4.3%
7	Paper	1,432	2.9%
8	Leather	1,132	2.3%
9	Glass	955	1.9%
10	Rubber	687	1.4%
11	Foam	168	0.3%
12	Snow	147	0.3%
13	Sand	58	0.1%
14	Other (32 materials)	1,731	3.4%

to create a comprehensive, diverse, and standardized collection of Physical-based assets annotated with 20+ views rendered images, physical properties, and corresponding guiding videos.

C.2. Material and Physical Property Distribution

The dataset exhibits rich material diversity, covering 46 distinct material categories. Among these, 14 primary materials constitute the majority of the dataset, while 32 additional rare materials provide coverage for specialized physical scenarios. Table 2 presents the distribution of the 14 primary materials.

The most represented material is **Plastic**, with 13,696 samples (27.3%), reflecting its prevalence in manufactured objects. **Wood** constitutes the second largest category with 8,443 samples (16.8%), followed by **Metal** with 7,353 samples (14.6%) and **Fabric** with 7,255 samples (14.5%). **Ceramic** objects account for 3,023 samples (6.0%). Medium-frequency materials include **Stone** (2,135 samples, 4.3%), **Paper** (1,432 samples, 2.9%), **Leather** (1,132 samples, 2.3%), **Glass** (955 samples, 1.9%), and **Rubber** (687 samples, 1.4%). Low-frequency but physically interesting materials comprise **Foam** (168 samples, 0.3%), **Snow** (147 samples, 0.3%), and **Sand** (58 samples, 0.1%). The remaining 32 rare materials collectively account for approximately 3.0% of the dataset, providing diversity for edge cases and specialized physical behaviors.

This heterogeneous material distribution enables our model to learn a comprehensive physical prior spanning rigid bodies (metal, stone), deformable materials (rubber, foam), granular substances (sand, snow), and everyday materials (plastic, wood, fabric). The long-tail distribution also facilitates studying generalization to rare material types.

Young’s Modulus (E): Measures material stiffness, ranging from soft materials (10^3 Pa) to rigid materials (4×10^{11} Pa). The dataset contains 10 distinct values spanning this range.

Poisson’s Ratio (ν): Characterizes material compressibility, typically ranging from 0.01 to 0.49. The dataset includes 10 representative values covering common material behaviors.

C.3. Source Datasets

Our dataset aggregates models from the following four sources, each contributing unique characteristics:

OmniObject3D A high-fidelity dataset featuring approximately 6,000 real-world scanned objects across 190 common categories (e.g., cups, chairs, animal models). It provides rich multi-modal data, including textured meshes with millimeter-level geometric accuracy and multi-view rendered images. For our purposes, we primarily leveraged its high-resolution rendered views (e.g., the 24-view set with associated camera parameters) to extract detailed appearance and geometric information.

HSSD Dataset. The Habitat Synthetic Scenes Dataset (HSSD) [8], contains over 18,000 high-quality indoor scenes with photorealistic rendering and detailed semantic annotations. The dataset features diverse residential and commercial environments with realistic layouts and furnishings.

Amazon Berkeley Objects (ABO) ABO offers a collection of approximately 8,000 high-quality, industry-standard 3D models covering 98 everyday object categories. The data includes textured CAD models (.obj/.glb), which we utilized to generate consistent multi-view renderings that align with our standardized format.

Objaverse A 10M+ dataset containing millions of 3D objects, Objaverse offers unparalleled diversity in object shape, category, and style. We selected a substantial subset from this collection to significantly broaden the scope and variety of our final dataset, as detailed in the following section.

C.4. Data Processing

Filter and Render To ensure quality and consistency across the heterogeneous source datasets, we established a systematic data curation and processing pipeline. The Objaverse dataset, while extensive, is characterized by its considerable size and variable data quality. Consequently, to extract a high-quality subset, we employed a systematic curation strategy analogous to the one applied to gobjaverse. The screening procedure is outlined as follows: (1) A geometric similarity clustering algorithm was employed

to identify and remove near-duplicate models. Any model exhibiting a similarity score of over 85% with another was considered redundant and removed; (2) To filter out objects with non-standard or incomplete textures, we performed an analysis in the HSV (Hue, Saturation, Value) color space. Models where white pixels constituted more than 75% of the surface texture were discarded, as this often indicates missing or placeholder textures. In the end, we filtered approximately about 20k data points in the Objaverse. For the other datasets, we used the full data without applying a filtering process. For datasets that do not provide enough view rendering view, we use the rendering script provided by the corresponding dataset for enough view rendering. This procedure ensures that every object in our dataset is represented by a consistent set of views, capturing its complete geometric features for subsequent learning tasks.

D. Dataset Construction Pipeline.

As shown in Figure 1, we construct PhysAssets through an automated pipeline which predicting physical properties (material class, Young’s modulus, Poisson’s ratio) from 8 selected views using Qwen3-VL [14] and generating ground-truth reference videos using FramePack [18] conditioned on predicted properties. This pipeline enables scalable annotation of 50,215 objects with physical properties and reference dynamics.

D.1. Physical Property Annotation Pipeline

We develop a semi-automatic annotation pipeline leveraging multimodal large language models to predict three critical physical properties for each object: material class, Young’s modulus (E), and Poisson’s ratio (ν). This approach enables scalable annotation of large-scale 3D datasets while maintaining consistency with real-world material physics.

D.1.1. Visual Feature Extraction

For each 3D object, we choose eight uniformly distributed views at fixed elevation. These multi-view RGB images provide comprehensive visual coverage of the object’s geometry, texture, and appearance. The views are then fed into Qwen3-VL [14], a state-of-the-art vision-language model pre-trained on diverse visual and textual data.

Material Classification Material classification is performed through vision-language alignment. We define a closed vocabulary of 45 primary materials commonly found in everyday objects: *Wood, Metal, Plastic, Glass, Fabric, Leather, Ceramic, Stone, Rubber, Paper, Sand, Snow, Plasticine, Foam, etc.*. The model is queried with the following prompt:

Material Classification Prompt:

“What is the primary material of the object in these images? Answer with a single word from this list: Wood, Metal, Plastic, Glass, Fabric, Leather, Ceramic, Stone, Rubber, Paper, Sand, Snow, Plasticine, Foam, etc.”

The model processes all eight views and outputs a material label based on cross-modal similarity between visual features and material descriptions. In cases of ambiguity, a weighted voting mechanism across views determines the final material class.

Young’s Modulus Prediction Young’s modulus (E) characterizes material stiffness—the resistance to elastic deformation under stress. We discretize the continuous range of Young’s modulus values into 10 interpretable categories spanning from extremely soft materials (e.g., gel, foam) to ultra-stiff materials (e.g., diamond, tungsten). The model is prompted with:

“Based on these images, determine the Young’s Modulus (E) of the object.

What is Young’s Modulus?

Young’s Modulus (E) measures a material’s stiffness or resistance to elastic deformation. It indicates how much stress is needed to produce a given amount of strain (deformation).

Select the most appropriate description:

1. *extremely soft - Like gel or foam (e.g., jelly, soft foam) ~ 1 KPa*
2. *very soft - Like rubber or sponge (e.g., rubber bands, foam mattress) ~ 100 KPa*
3. *soft - Like soft plastics or leather (e.g., leather, soft PVC) ~ 1 MPa*
4. *moderately soft - Like hard rubber (e.g., tire rubber) ~ 10 MPa*
5. *moderate - Like nylon or cork (e.g., nylon, wood cork) ~ 100 MPa*
6. *moderately stiff - Like hard plastics (e.g., ABS plastic, acrylic) ~ 1 GPa*
7. *stiff - Like glass or ceramics (e.g., glass, porcelain) ~ 10 GPa*
8. *very stiff - Like aluminum (e.g., aluminum, brass) ~ 70 GPa*
9. *extremely stiff - Like steel (e.g., steel, iron) ~ 200 GPa*
10. *ultra stiff - Like tungsten or diamond (e.g., tungsten, diamond) ~ 400 GPa*

Answer with ONLY ONE of these exact keywords.”

The predicted categorical label is then mapped to a numerical value in Pascals (Pa) using the mapping defined in Table 3.

Poisson’s Ratio Prediction Poisson’s ratio (ν) quantifies the ratio of lateral strain to axial strain when a material is

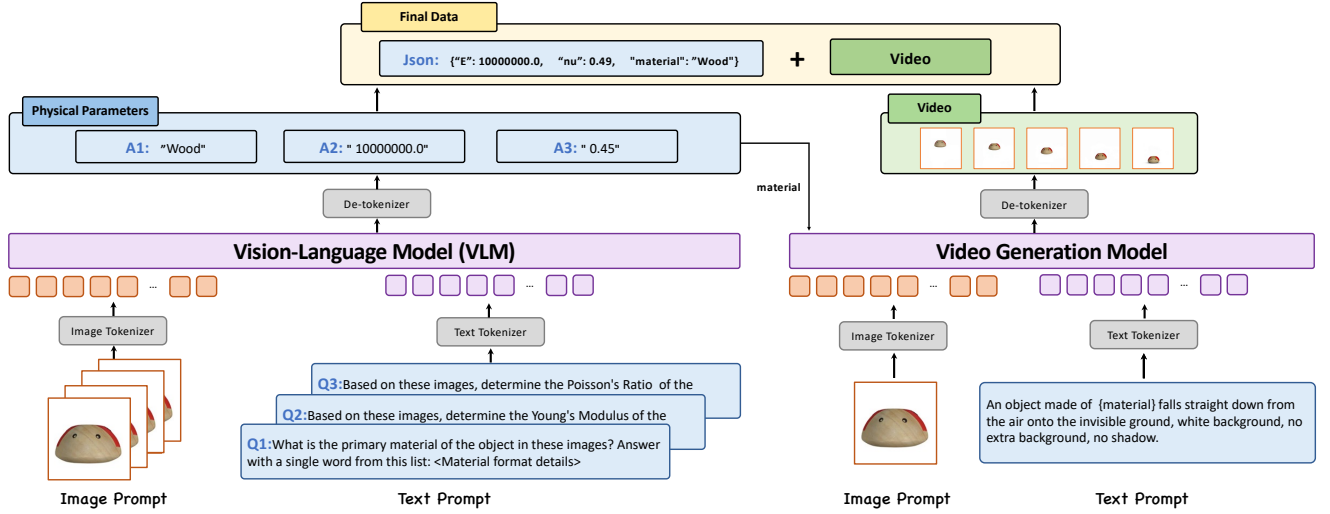


Figure 1. Automated dataset construction pipeline. We predict physical properties using Qwen3-VL, and generate reference videos using FramePack.

deformed. We similarly discretize Poisson’s ratio into 10 categories representing different material behaviors, from auxetic materials (negative Poisson’s ratio) to nearly incompressible materials (approaching 0.5). The prediction prompt is:

Poisson’s Ratio Prediction Prompt:

“Based on these images, determine the Poisson’s Ratio (ν) of the object.

What is Poisson’s Ratio?

Poisson’s Ratio (ν) measures how much a material expands laterally when compressed axially, or contracts laterally when stretched. It describes the relationship between lateral strain and axial strain.

Select the most appropriate description:

1. nearly incompressible - Almost no volume change (e.g., rubber) ~ 0.50
2. high resistance - High lateral expansion (e.g., soft rubber) ~ 0.45
3. moderately high - Moderately high deformation (e.g., gold, lead) ~ 0.40
4. moderate high - Above average deformation (e.g., plastic, aluminum) ~ 0.35
5. moderate - Typical for many metals (e.g., steel, iron) ~ 0.30
6. moderate low - Below average deformation (e.g., glass) ~ 0.25
7. low - Low lateral expansion (e.g., concrete, ceramics) ~ 0.20
8. very low - Very low lateral expansion (e.g., cork) ~ 0.15
9. extremely low - Minimal lateral deformation (e.g., foam) ~ 0.10

10. auxetic - Negative Poisson’s ratio materials ~ 0.01

Answer with **ONLY ONE** of these exact keywords.”

The categorical output is converted to a dimensionless numerical value using the mapping in Table 4.

Property Mapping Tables The categorical predictions from the vision-language model are mapped to numerical physical property values suitable for Material Point Method (MPM) simulation. Tables 3 and 4 present the complete mappings.

Table 3. Young’s Modulus categorical to numerical mapping. Values span 8 orders of magnitude, covering materials from soft gels to ultra-hard ceramics.

Category	Example Materials	Value (Pa)
extremely soft	Gel, foam, jelly	1.0×10^3
very soft	Rubber, sponge, silicone	1.0×10^5
soft	Leather, soft PVC, fabric	1.0×10^6
moderately soft	Hard rubber, tire rubber	1.0×10^7
moderate	Nylon, cork, paper	1.0×10^8
moderately stiff	Hard plastic (ABS, acrylic)	1.0×10^9
stiff	Glass, ceramic, porcelain	1.0×10^{10}
very stiff	Aluminum, brass, bronze	7.0×10^{10}
extremely stiff	Steel, iron, stainless steel	2.0×10^{11}
ultra stiff	Tungsten, diamond, carbide	4.0×10^{11}

D.2. Video Generation and Preference Calculation

To facilitate the second stage of our training, which employs Direct Preference Optimization (DPO), we established a systematic pipeline for generating a dataset of preference

Table 4. Poisson’s Ratio categorical to numerical mapping. Values range from 0.01 (auxetic materials) to 0.49 (nearly incompressible materials).

Category	Example Materials	Value
auxetic	Special engineered materials	0.01
extremely low	Foam materials	0.10
very low	Cork, engineered materials	0.15
low	Concrete, ceramics, brick	0.20
moderate low	Glass, cast iron	0.25
moderate	Steel, iron, brass, titanium	0.30
moderate high	Plastic, aluminum, copper	0.35
moderately high	Gold, lead, clay	0.40
high resistance	Soft rubber, flexible polymers	0.45
nearly incompressible	Rubber, elastomers	0.49

tuples. This process is crucial for providing the high-quality, ranked data required to fine-tune our model on the nuances of physical dynamics. The pipeline consists of three main steps:

D.2.1. Ground-Truth Video Generation

We generate reference videos using FramePack [18], guided by text prompts describing the physical scenario. After evaluating multiple prompt formulations (detailed below), we selected the following template for its optimal balance of simplicity and physical realism:

“An object made of {material} falls straight down from the air onto the invisible ground, white background, no extra background, no shadow.”

D.2.2. Alternative Prompt Variants

For reference and reproducibility, we document the alternative prompt variants explored during our experimentation. These prompts represent different trade-offs between prompt complexity, physical constraints, and generation control.

Prompt 2 (Detailed Physics Description):

“A {material} toy centered on a plain pure white background. The {material} toy falls straight down vertically from the center of the frame to the bottom edge, obeying the laws of physics (gravity, acceleration). Show the entire descent: starting stationary at center, accelerating downwards, hitting the bottom edge with a subtle impact, and coming to a complete stop. The {material} toy remains rigid and inanimate throughout, showing no deformation or independent movement. Fixed, static camera view. No anthropomorphism, no unexpected motion, only the physics-based vertical fall and stop.”

Limitation: Overly detailed constraints sometimes led to inconsistent generation or failure to satisfy all specified conditions.

Prompt 5 (Identity Preservation Focus):

“Generate a short, high-fidelity video based on the provided object image, where the absolute highest priority is to strictly maintain the object’s identity throughout the entire sequence. The scene features a seamless white background and a solid, invisible, horizontal white floor. The video begins with the object perfectly still in mid-air; then it is released to fall straight down vertically under gravity. Crucially, the object must maintain its initial orientation during the fall, without any tumbling, spinning, or rotation. The object is made of {material}, and its impact and subsequent behavior must realistically simulate the physical properties of this material.”

Limitation: While improving identity preservation, this prompt occasionally resulted in unrealistic motion due to strict orientation constraints.

Prompt 6 (Photorealism Emphasis):

“Generate a short, photorealistic video based on the provided input image, simulating the object falling and impacting the ground. Throughout the entire video, the object must retain its original visual identity—its shape, texture, and color. The fall itself must be completely inanimate and passive; the object must descend in a pure vertical drop without any rotation, spinning, or tumbling. Upon impact with the flat white ground, the object’s physical reaction must precisely mimic the properties of {material}. The entire event takes place in a seamless, infinite white studio environment.”

Limitation: We found that excessively long prompts with detailed constraints often compromise generation quality, leading to inconsistent or unnatural motion.

The selected prompt consistently produced the most physically plausible and visually coherent videos across diverse materials and object geometries.

D.2.3. Candidate Video Generation

Leveraging the model pre-trained in Stage 1, we generate a set of plausible, yet varied, simulation outcomes. Specifically, we sample three distinct sets of physical properties (e.g., Young’s modulus, Poisson’s ratio) from the learned probability distribution associated with the object. Each of these property sets is then used to run a new simulation, producing three unique candidate videos that represent different potential physical behaviors.

D.2.4. Preference Labeling via Trajectory Alignment

To create preference pairs for DPO training, we develop an automatic labeling pipeline that compares simulated dynamics against reference videos through three-stage trajectory alignment. We employ SAM-2 [11] for object segmentation and CoTracker-3 [7] for dense trajectory extraction across both ground-truth and simulated sequences.

Spatial Alignment. Due to different camera viewpoints and object scales between reference and simulated videos, direct trajectory comparison is infeasible. We address this through bounding box normalization: for each video, we compute the object’s bounding box from its segmentation mask as $\mathcal{B} = (x_{\min}, y_{\min}, x_{\max}, y_{\max})$. Point trajectories from the ground-truth video are first normalized to $[0, 1]$ coordinates relative to its bounding box:

$$\mathbf{p}^{\text{norm}} = \left(\frac{x - x_{\min}}{x_{\max} - x_{\min}}, \frac{y - y_{\min}}{y_{\max} - y_{\min}} \right) \quad (5)$$

These normalized coordinates are then mapped to the simulated video’s coordinate frame using its bounding box parameters. This spatial alignment ensures correspondence between trajectories regardless of viewpoint or scale differences.

Landing Frame Alignment. Physical simulations may exhibit different temporal dynamics (e.g., falling speeds) even with similar physical properties. To enable fair comparison, we align sequences based on a key physical event: the object’s landing moment. Specifically, we identify the landing frame as the temporal turning point where the object’s vertical motion reverses. For each video, we track the point with maximum y -coordinate in the first frame (typically the object’s bottom) and monitor its trajectory. The landing frame f^* is detected when the vertical velocity changes sign:

$$f^* = \arg \min_t \{t \mid y_t \leq y_{t-1}, t > 0\} \quad (6)$$

where y_t represents the tracked point’s y -coordinate at frame t . This frame marks the transition from falling to resting/bouncing phases.

Temporal Alignment. Using the detected landing frames $(f_{\text{GT}}^*, f_{\text{sim}}^*)$ as temporal anchors, we align the post-landing phases of both sequences. We determine the comparable duration as $T = \min(T_{\text{GT}} - f_{\text{GT}}^*, T_{\text{sim}} - f_{\text{sim}}^*)$, where T_{GT} and T_{sim} are the total frame counts. Additionally, we compute a spatial offset $(\Delta x, \Delta y)$ between the landing positions in both videos and apply this correction to the simulated trajectories:

$$\mathbf{p}_{\text{sim}}^{\text{aligned}} = \mathbf{p}_{\text{sim}} + (\Delta x, \Delta y) \quad (7)$$

This ensures that both sequences are aligned not only temporally but also spatially at the critical landing event.

Similarity Metric. After three-stage alignment, we compute the trajectory dissimilarity as:

$$d(\mathcal{V}_{\text{sim}}, \mathcal{V}_{\text{GT}}) = \frac{1}{NT} \sum_{n=1}^N \sum_{t=1}^T \|\mathbf{p}_{n,t}^{\text{GT}} - \mathbf{p}_{n,t}^{\text{sim}}\|_2 \quad (8)$$

where N is the number of tracked points and T is the aligned sequence length. Lower dissimilarity indicates better physical plausibility. For each scene, we rank K candidate simulations by this metric and select the best as the “winner” and worst as the “loser” for DPO training.

D.2.5. Additional Data Sources

It is also worth noting that the PhysX [1] dataset was released concurrently with our research, offering 3D objects annotated with physical properties, which is also suitable for our dataset process. Given the timing constraints, its integration was not feasible for the present study. Nevertheless, we acknowledge its significance and view it as a promising avenue for extending our work in the future.



Figure 2. **Multi-view generation using MVAdapter.** Given a single frontal view image as input (left), MVAdapter [6] generates three auxiliary views: rear, left, and right (right three panels). These synthesized views, together with the input frontal view, provide comprehensive angular coverage for our 3D Gaussian reconstruction and physics prediction pipeline. The generated views maintain consistent geometry and appearance while capturing different perspectives of the object.

E. Additional Results

To fully demonstrate the versatility and effectiveness of our approach, we present an extended suite of supplementary experiments with comprehensive qualitative insights. Specifically, Figure 3 provides detailed visualizations of the Multi-View Stereo (MVS) module, showcasing its exceptional ability to accurately reconstruct 3D geometry from multi-view inputs. Given four randomly selected input

views, the model generates novel viewpoints, and we visualize four representative views sampled from eight randomly selected output views as examples. Complementarily, Figure 2 offers visualizations of the MVAdapter component, clearly revealing how it effectively bridges domain gaps and enhances feature alignment across diverse input modalities. Beyond the core component validations, Figure 4 and Figure 5 exhibit the model’s performance on fundamental stretching and dropping scenarios, respectively. We further push the envelope to validate its effectiveness under more challenging configurations: Figure 6 illustrates strong robustness in cluttered/complex background scenes, while Figure 7 highlights its superior capability in handling intricate multi-object interactions and other results in Figure 8 9 10.

F. Limitations and Future Work

While PhysGM demonstrates significant advances in fast, physically-grounded 3D synthesis, it is important to acknowledge its current limitations, which also highlight promising directions for future research.

Data Dependency and Generalization. Our model’s performance is inherently tied to the scope and diversity of the PhysAssets dataset. While large, the dataset primarily consists of rigid objects. Consequently, the model may not generalize well to out-of-distribution categories, such as highly deformable or articulated objects. Future work could focus on expanding the dataset and exploring domain adaptation techniques to handle a wider variety of object types.

Simplified Physics Representation. PhysGM currently predicts a single, “lumped” vector of physical properties (e.g., one mass, one friction coefficient) for the entire object. This assumes uniform material composition, which is not true for many real-world objects (e.g., a hammer with a metal head and wooden handle). A significant next step would be to extend our framework to predict spatially varying material properties, enabling more complex and realistic simulations.

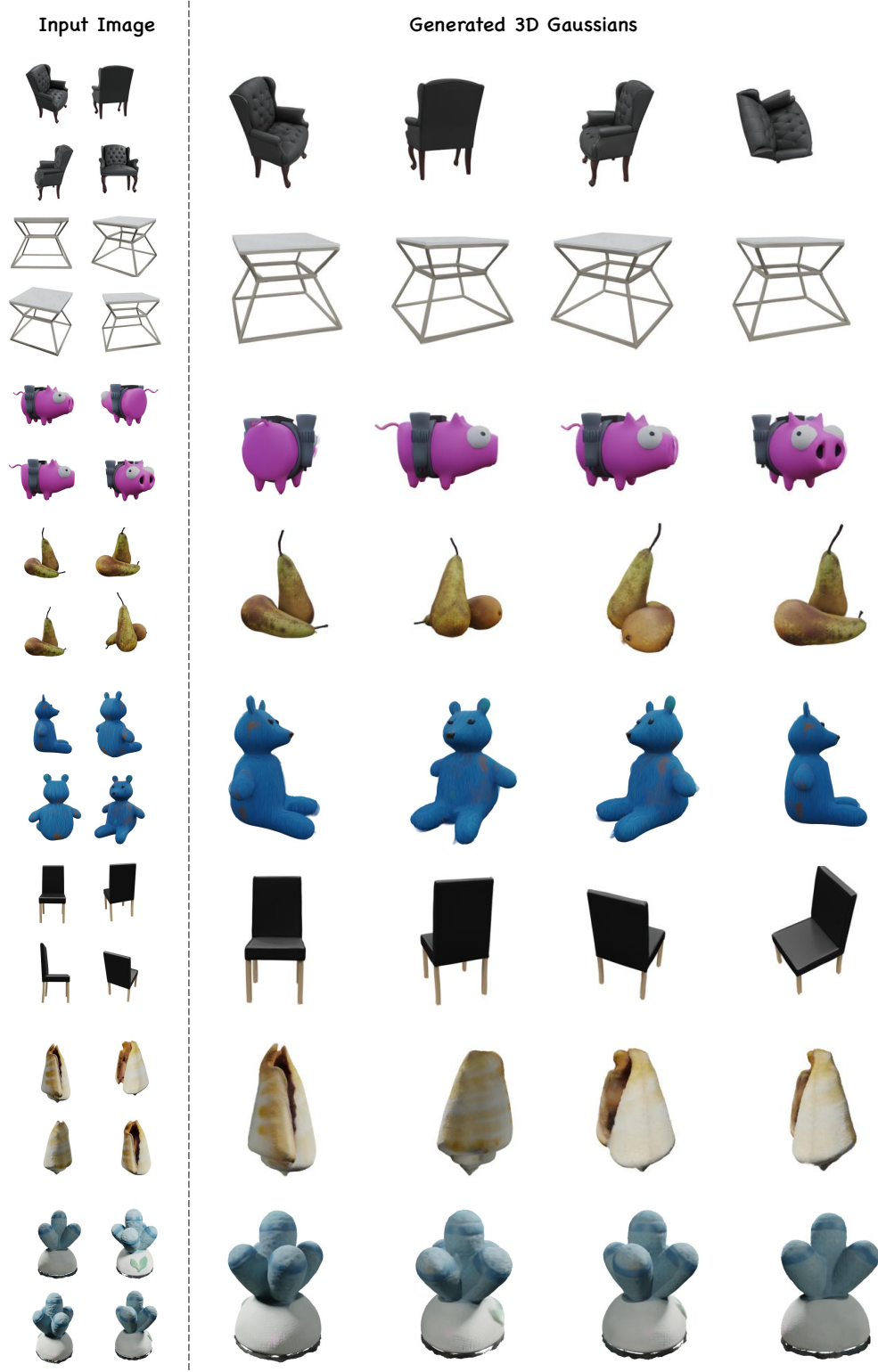


Figure 3. Qualitative results for Multi-View Stereo. Our method generates Gaussian splatting with remarkable visual quality on various challenging images.



Figure 4. Qualitative results for stretching scenarios. Our method correctly captures the distinct responses of different materials under tensile forces.

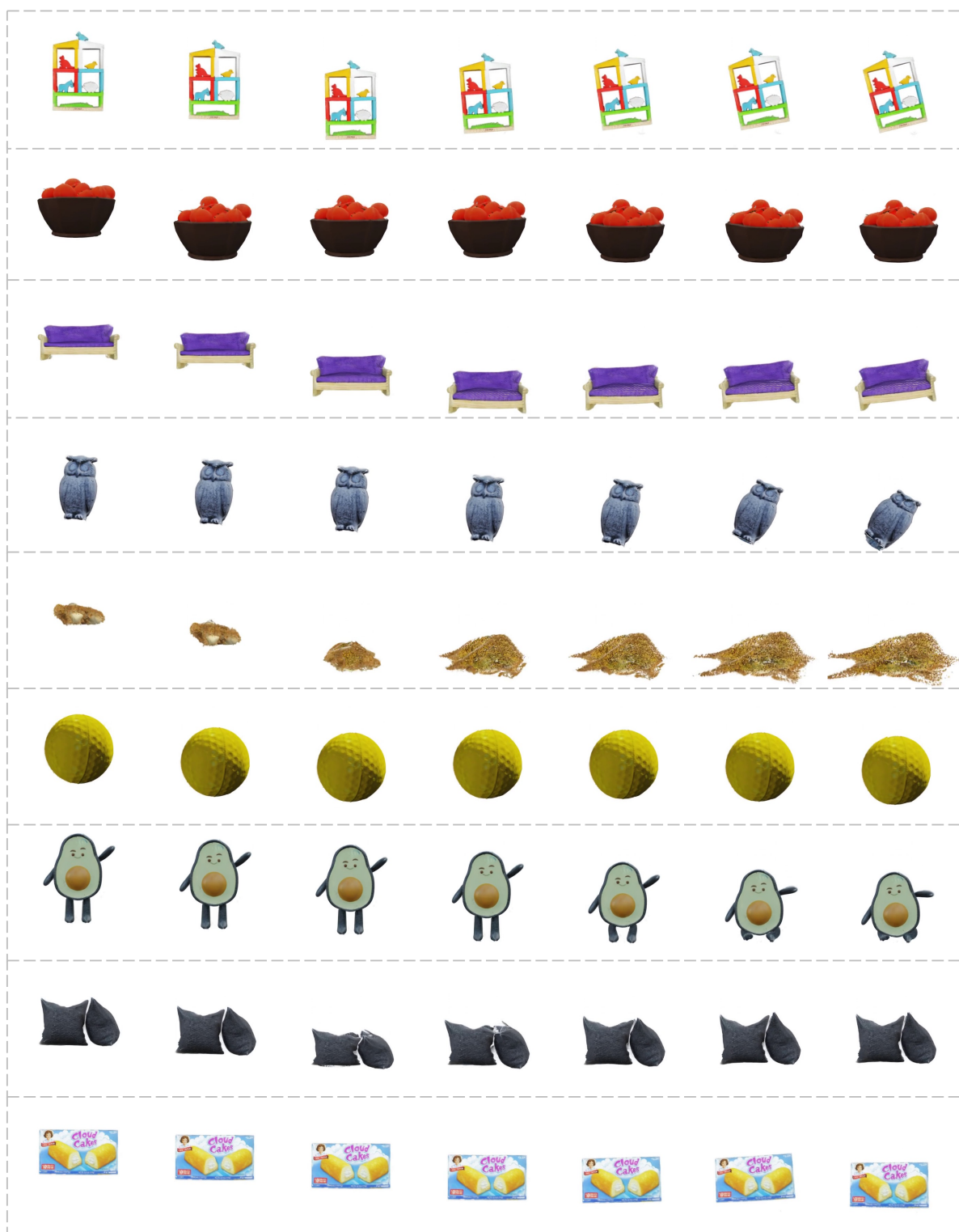


Figure 5. Qualitative results for dropping scenarios. Our model accurately predicts the physical properties of different materials, leading to plausible deformation and final states upon impact with the ground.



Figure 6. Demonstration of our model’s robustness in in-the-wild scenes.



Figure 7. Qualitative results for multi-object interaction scenarios. Our approach can handle more complex scenes involving simultaneous collisions and interactions, generating physically consistent outcomes for all objects.

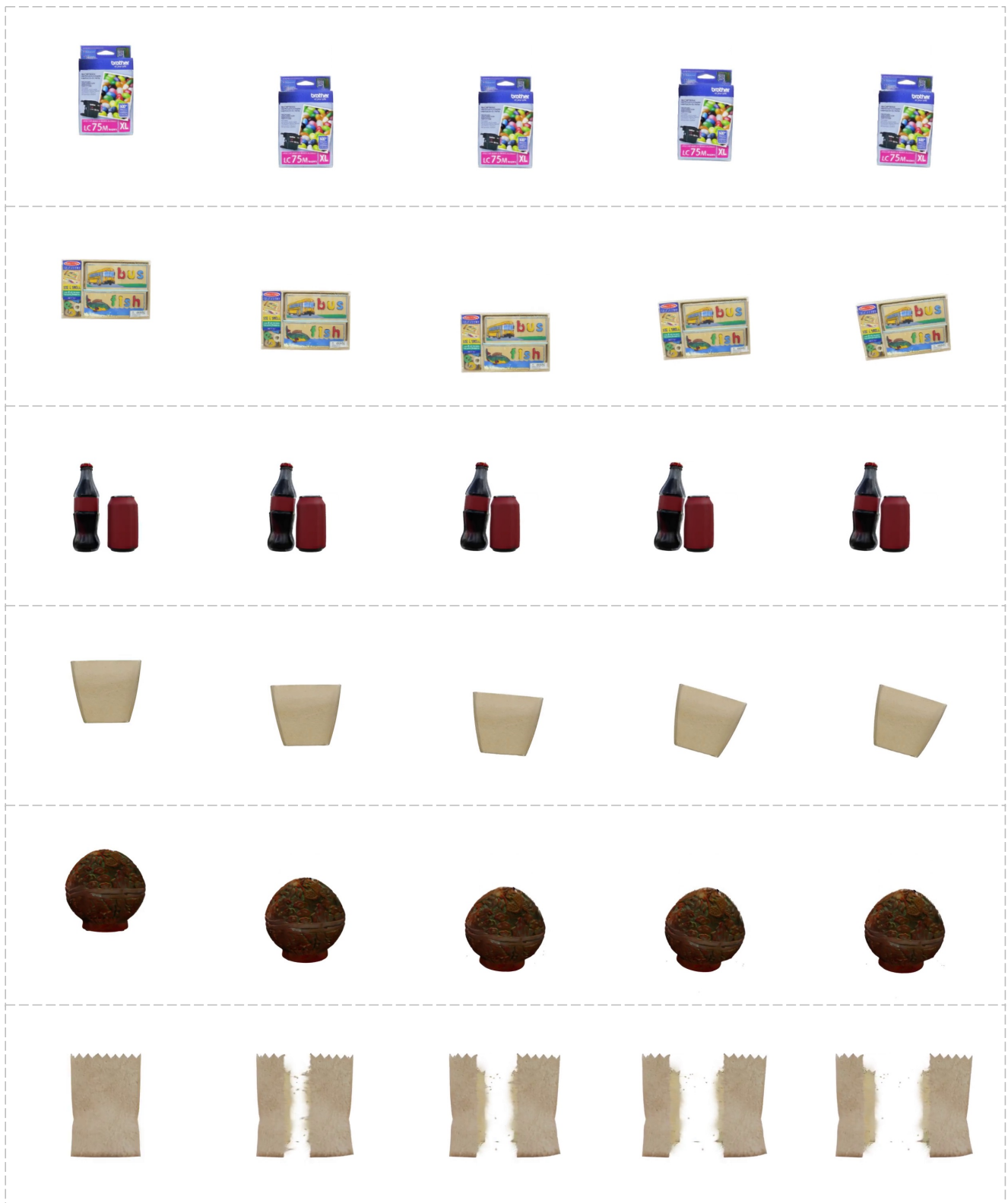


Figure 8. Other results.

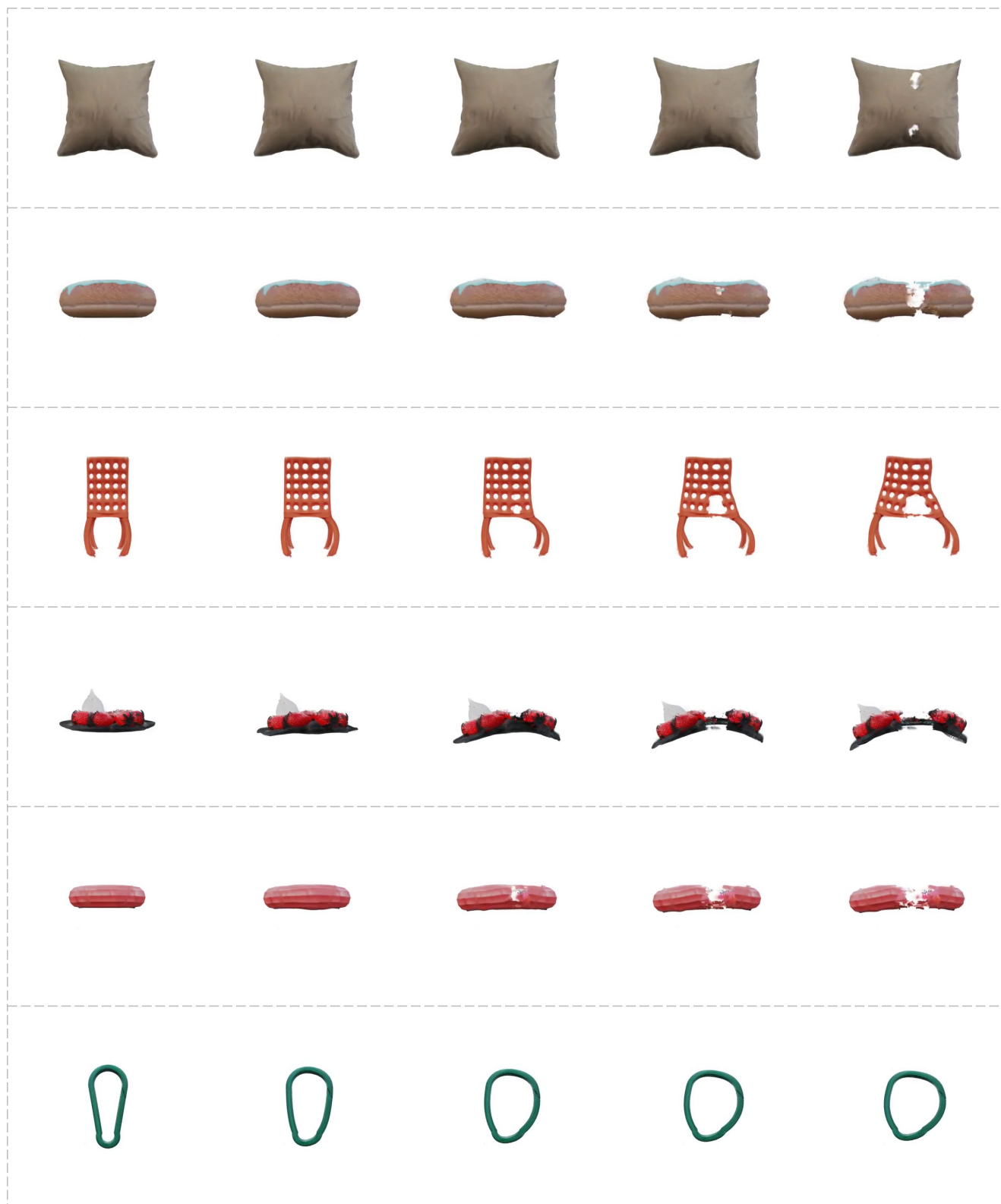


Figure 9. Other results.

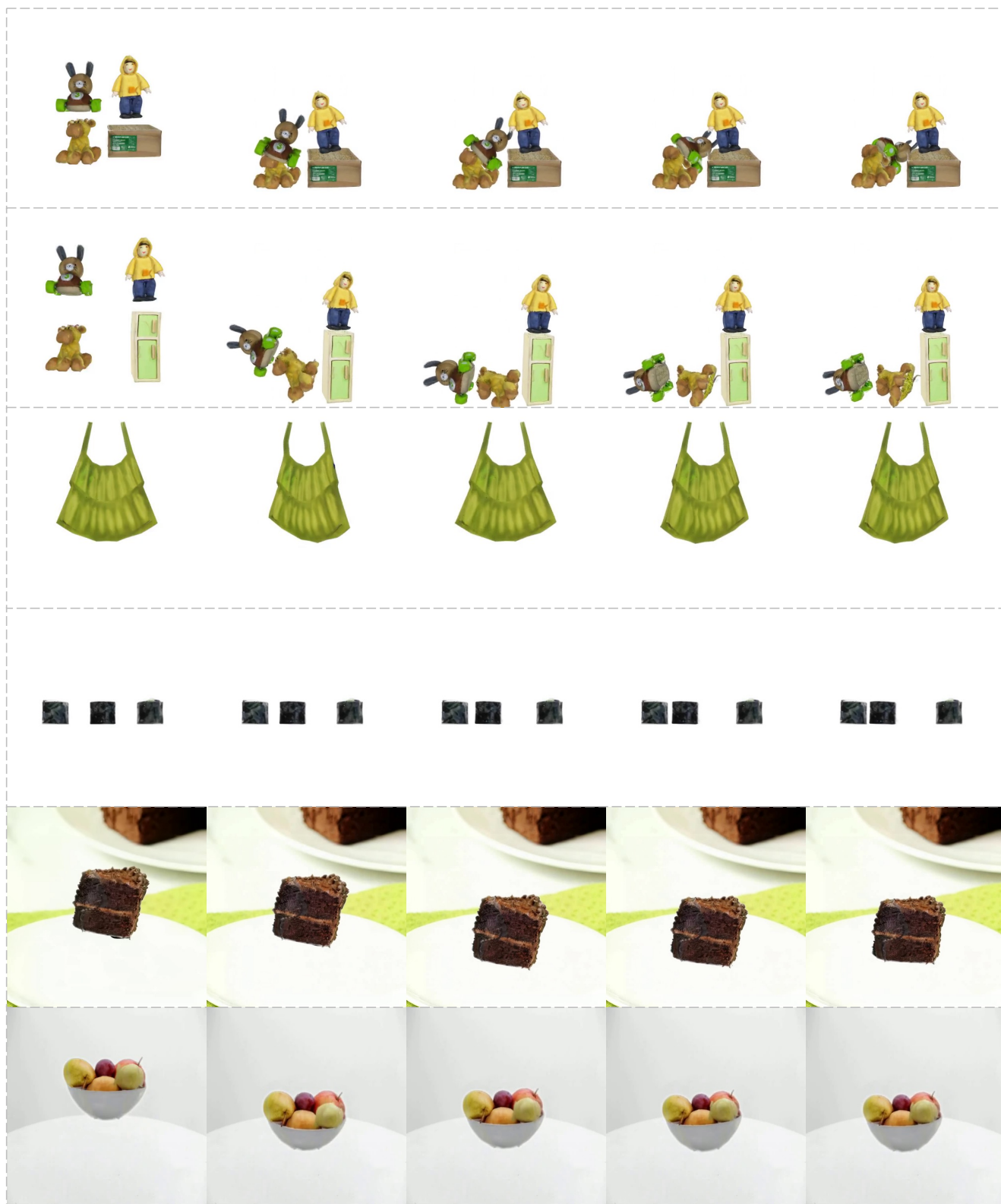


Figure 10. Other results.

References

- [1] Ziang Cao, Zhaoxi Chen, Linag Pan, and Ziwei Liu. Physx: Physical-grounded 3d asset generation. *arXiv preprint arXiv:2507.12465*, 2025. 8
- [2] Jasmine Collins, Shubham Goel, Kenan Deng, Achleshwar Luthra, Leon Xu, Erhan Gundogdu, Xi Zhang, Tomas F Yago Vicente, Thomas Dideriksen, Himanshu Arora, et al. Abo: Dataset and benchmarks for real-world 3d object understanding. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 21126–21136, 2022. 3
- [3] Matt Deitke, Dustin Schwenk, Jordi Salvador, Luca Weihs, Oscar Michel, Eli VanderBilt, Ludwig Schmidt, Kiana Ehsani, Aniruddha Kembhavi, and Ali Farhadi. Objaverse: A universe of annotated 3d objects. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 13142–13153, 2023. 3
- [4] Laura Downs, Anthony Francis, Nate Koenig, Brandon Kinman, Ryan Hickman, Krista Reymann, Thomas B McHugh, and Vincent Vanhoucke. Google scanned objects: A high-quality dataset of 3d scanned household items. In *2022 International Conference on Robotics and Automation (ICRA)*, pages 2553–2560. IEEE, 2022. 1, 3
- [5] Tianyu Huang, Haoze Zhang, Yihan Zeng, Zhilu Zhang, Hui Li, Wangmeng Zuo, and Rynson WH Lau. Dreamphysics: Learning physics-based 3d dynamics with video diffusion priors. In *Proceedings of the AAAI Conference on Artificial Intelligence*, pages 3733–3741, 2025. 1, 2
- [6] Zehuan Huang, Yuan-Chen Guo, Haoran Wang, Ran Yi, Lizhuang Ma, Yan-Pei Cao, and Lu Sheng. Mv-adapter: Multi-view consistent image generation made easy. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 16377–16387, 2025. 8
- [7] Nikita Karaev, Iurii Makarov, Jianyuan Wang, Natalia Neverova, Andrea Vedaldi, and Christian Rupprecht. Co-tracker3: Simpler and better point tracking by pseudo-labelling real videos. In *Proc. arXiv:2410.11831*, 2024. 8
- [8] Mukul Khanna*, Yongsan Mao*, Hanxiao Jiang, Sanjay Haresh, Brennan Shacklett, Dhruv Batra, Alexander Clegg, Eric Undersander, Angel X. Chang, and Manolis Savva. Habitat Synthetic Scenes Dataset (HSSD-200): An Analysis of 3D Scene Scale and Realism Tradeoffs for ObjectGoal Navigation. *arXiv preprint*, 2023. 3, 4
- [9] Yuchen Lin, Chenguo Lin, Jianjin Xu, and Yadong Mu. Omniphysgs: 3d constitutive gaussians for general physics-based dynamics generation. *arXiv preprint arXiv:2501.18982*, 2025. 1, 2
- [10] Ri-Zhao Qiu, Ge Yang, Weijia Zeng, and Xiaolong Wang. Feature splatting: Language-driven physics-based scene synthesis and editing. *arXiv preprint arXiv:2404.01223*, 2024. 1, 2
- [11] Nikhila Ravi, Valentin Gabeur, Yuan-Ting Hu, Ronghang Hu, Chaitanya Ryali, Tengyu Ma, Haitham Khedr, Roman Rädle, Chloe Rolland, Laura Gustafson, et al. Sam 2: Segment anything in images and videos. *arXiv preprint arXiv:2408.00714*, 2024. 8
- [12] Jiawei Ren, Liang Pan, Jiaxiang Tang, Chi Zhang, Ang Cao, Gang Zeng, and Ziwei Liu. Dreamgaussian4d: Generative 4d gaussian splatting. *arXiv preprint arXiv:2312.17142*, 2023. 1, 2
- [13] Oriane Siméoni, Huy V. Vo, Maximilian Seitzer, Federico Baldassarre, Maxime Oquab, Cijo Jose, Vasil Khalidov, Marc Szafraniec, Seungeun Yi, Michaël Ramamonjisoa, Francisco Massa, Daniel Haziza, Luca Wehrstedt, Jianyuan Wang, Timothée Darcet, Théo Moutakanni, Leonel Sentana, Claire Roberts, Andrea Vedaldi, Jamie Tolan, John Brandt, Camille Couprie, Julien Mairal, Hervé Jégou, Patrick Labatut, and Piotr Bojanowski. Dinov3, 2025. 1
- [14] Qwen Team. Qwen3 technical report, 2025. 5
- [15] Zhou Wang, Alan C Bovik, Hamid R Sheikh, and Eero P Simoncelli. Image quality assessment: from error visibility to structural similarity. *IEEE transactions on image processing*, 13(4):600–612, 2004. 1
- [16] Tong Wu, Jiarui Zhang, Xiao Fu, Yuxin Wang, Jiawei Ren, Liang Pan, Wayne Wu, Lei Yang, Jiaqi Wang, Chen Qian, et al. Omniobject3d: Large-vocabulary 3d object dataset for realistic perception, reconstruction and generation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 803–814, 2023. 3
- [17] 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. 1, 2
- [18] Lvmin Zhang and Maneesh Agrawala. Packing input frame context in next-frame prediction models for video generation. *arXiv preprint arXiv:2504.12626*, 2025. 5, 7
- [19] Richard Zhang, Phillip Isola, Alexei A Efros, Eli Shechtman, and Oliver Wang. The unreasonable effectiveness of deep features as a perceptual metric. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 586–595, 2018. 1
- [20] Tianyuan Zhang, Hong-Xing Yu, Rundi Wu, Brandon Y Feng, Changxi Zheng, Noah Snively, 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, 2024. 1, 2
- [21] Haoyu Zhao, Hao Wang, Xingyue Zhao, Hao Fei, Hongqiu Wang, Chengjiang Long, and Hua Zou. Efficient physics simulation for 3d scenes via mllm-guided gaussian splatting. *arXiv preprint arXiv:2411.12789*, 2024. 1, 2