AutoPhyX: Automatic Text-Condition Physics Property Generation

Anonymous ECCV 2026 Submission

Paper ID #7088

Abstract. High-fidelity 4D physical simulation requires accurate physical parameter assignments that align with diverse material properties. Previous methods struggle with critical bottlenecks: they either rely on computationally intensive per-scene optimization or predict parameters from purely visual inputs. The latter suffers from inherent ambiguity, *e.g.*, the inability to distinguish the stiffness of rubber from its appearance alone. To bridge this gap, we propose **AutoPhyX**, a text-conditioned framework for predicting spatially-varying physical properties. Since training robust feed-forward models requires fine-grained supervision, we first introduce a novel part-controllable data generation pipeline that decomposes complex 3D assets into semantically distinct components and pairs them with physically plausible, diverse parameters and corresponding text descriptions. Leveraging this dataset, **AutoPhyX** employs a cross-modal modulation mechanism where text features dynamically modulate visual features for precise physical grounding. By formulating property prediction within a voxel field, **AutoPhyX** ensures compatibility with diverse 3D representations, including meshes, point clouds, Gaussian Splatting, and NeRF. Experiments demonstrate that our method enables physically plausible, text-driven parameter assignment in a single forward pass. It achieves high accuracy and robust generalization to in-the-wild objects, paving the way for future physics-intensive embodied and robotic applications.

Keywords: Physics Simulation · Parameter Prediction · Cross-modal Modulation

1 Introduction

High-fidelity 4D physical simulation [27, 36, 40, 48] has become a vital foundation for interactive digital environments, driving applications ranging from Finite Element Method (FEM) based [2, 4–6, 31] robotic policy training, to Smoothed Particle Hydrodynamics (SPH) driven [26,32–35,39,43] complex fluid and glacier calving simulations, and Material Point Method (MPM) powered [3, 9, 10, 14, 19, 21,42,46,46,47] real-time virtual reality (VR) experiences. Despite their ability to synthesize high-fidelity, realistic dynamics across diverse scenes, these physics-based approaches share a major bottleneck: they rely heavily on the manual assignment of physical parameters. This reliance on trial-and-error parameter

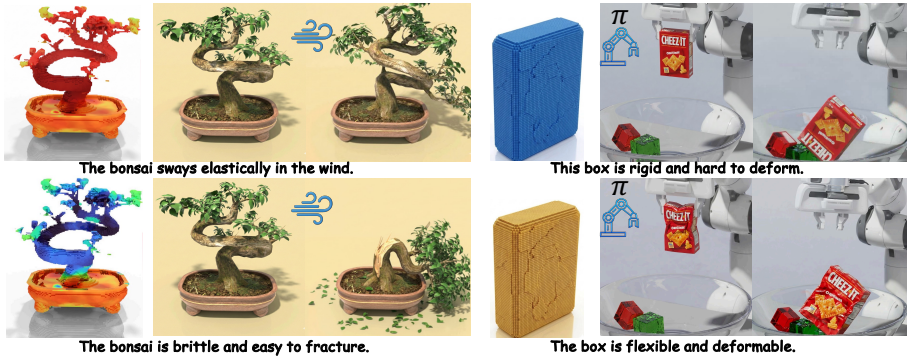


Fig. 1: Text-driven physical simulation with AutoPhyX. Our framework predicts dense physical parameters for 3D assets conditioned on textual descriptions. For each scenario, the leftmost images visualize the predicted voxelized physical parameters, which are normalized and mapped to RGB channels, corresponding to the input text. Under identical boundary conditions, the resulting simulations exhibit distinct behaviors, *e.g.*, the bonsai responding differently to the same wind force in MPM, and the box with different deformation under the same robot grasping policy in FEM.

design severely limits their scalability and applicability to complex scenes characterized by diverse, spatially varying materials.

Existing methods aiming to automate material acquisition primarily fall into two categories, each with its own limitations. Optimization-based methods [17, 25, 51, 54] leverage differentiable physics to iteratively refine material fields to match target observations. However, inferring particle-wise parameters from sparse supervision is ill-posed and computationally expensive, often requiring hours of per-scene optimization [25] and offering limited generalization across scenes. In contrast, feed-forward approaches such as Pixie [22] and VoMP [13] enable much faster inference; yet they rely purely on visual cues, which sacrifices semantic controllability and suffers from appearance ambiguity. For example, as shown in Figure 1, a tree could feature flexible branches that sway in the wind or brittle wood that fractures easily, while a biscuit box could be rigid cardboard or thin paper. Without explicit semantic guidance, it is difficult to disambiguate objects that are physically distinct but visually similar.

To address these critical limitations, we propose **AutoPhyX**, a feed-forward framework for fast, generalizable, and semantically controllable prediction of physical properties. Given a 3D asset and textual descriptions of its physical properties, the model predicts spatially dense physical parameters in a unified voxel field representation. This formulation provides a common physical abstraction that can be directly transferred to diverse downstream 3D formats, including Gaussian Splatting (GS), meshes, and point clouds, without per-scene optimization. To resolve the material-level ambiguity and capture spatially varying properties across multi-part, multi-material assets, **AutoPhyX** modulates the intermediate visual representation with a feature-wise affine transformation conditioned on the encoded text features, coupling high-level semantic cues with

fine-grained spatial representations. This cross-modal design enables the model to distinguish visually similar but physically different materials, while also assigning spatially coherent physical parameters to semantically distinct parts of the 3D asset.

Training such a robust, spatially-grounded model requires large-scale fine-grained supervision over multi-part, multi-material 3D assets with text descriptions, which is currently unavailable. To bridge this gap, we introduce an automated, part-controllable data generation pipeline. For each asset, we construct a 3D OpenCLIP [18] voxel representation, and leverage authoritative engineering databases (*e.g.*, MatWeb and Wikipedia) and Vision-Language Models (VLMs) to extract robust text-guided part segmentation, assign plausible material properties, and generate diverse part-aware textual descriptions. Our dataset, **Text2PhyX**, features 1,700 high-quality 3D assets, each paired with 8 distinct physical-parameter and text-description annotations, supplementing as training and testing data for text-conditioned physical property assignment.

We evaluate **AutoPhyX** extensively on **Text2PhyX** and the real-world ABO-500 [11,51] dataset, achieving state-of-the-art quantitative results on both. Specifically, our method improves the PSNR from 23.421 to 26.012 on our dataset and reduces the Average Displacement Error (ADE) by 20% compared to the second-best method on ABO-500. Furthermore, our predicted properties support a wide range of physical simulators, which we demonstrate through extensive results using both MPM and FEM. This versatility allows us to flexibly choose rendering and simulation pipelines on demand, *e.g.*, GS for real-time dynamic rendering, mesh in Blender for complex lighting or Isaac Gym for robotic policy training.

In summary, our main contributions are:

- **The first text-conditioned feed-forward framework for 3D physics prediction.** We introduce **AutoPhyX** that achieves precise semantic and physical grounding through a novel cross-modal modulation architecture, resolving the inherent ambiguity of previous pure-vision methods.
- **A large-scale, multi-part, multi-level 3D physics dataset.** We build **Text2PhyX** with dense part-aware physical annotations and paired text descriptions, enabling supervised learning for text-conditioned property prediction with an automated VLM-driven generation pipeline.
- **Extensive validation across diverse workflows.** We demonstrate that **AutoPhyX** significantly outperforms existing baselines in physical accuracy. More importantly, our approach bridges the gap between static 3D assets and various physics solvers, providing a plug-and-play solution that generalizes across varied representations, platforms, and rendering engines.

2 Related Work

Physical simulation has long been a cornerstone of computer graphics . Typical simulation methods, *e.g.*, FEM, which models stress-strain responses for rigid and soft bodies, and MPM [14,19], particularly suited for large deformations and

complex materials, have been widely adopted. Despite their success, they all rely on accurate physical parameters as input. In this work, we are particularly interested in three canonical properties: Young's modulus E, Poisson's ratio ν, and density ρ, because they jointly govern a material's stiffness, lateral deformation behavior, and inertial response, making them fundamental for a broad range of solid and deformable simulations. To eliminate the need for manual parameter assignment, prior efforts can be broadly grouped into the following directions.

2.1 Physics Property Assignment via Vision-Language Model

Prior research, such as NeRF2Physics [51], directly prompts VLM to predict exact physical parameters from images [7, 8, 16, 51]. However, this direct approach struggles for three main reasons: VLM lacks accurate built-in physics knowledge, their outputs are highly random, and materials suffer from severe visual ambiguity, where objects that look identical can have completely different physical properties. To overcome these issues, we shift the VLM's role from a direct physics calculator to a semantic reasoning engine supported by external tools. Specifically, instead of relying on the VLM to guess exact numbers, we give specific material requirements, *e.g.*, more deformable or crisp, and prompt it to query external databases to find reasonable physical ranges. To handle the randomness of VLM outputs, we query the model multiple times and select the best result. Finally, we use the VLM to generate text descriptions that explicitly define the material, effectively solving the visual ambiguity problem.

2.2 Test-time Physics Property Optimization

Another prominent line of research focuses on test-time optimization, which iteratively refines physical parameters to align simulations with target observations. Observation-based optimization techniques [20,20,23,24,28,37,50,52,53,55] typically minimize the discrepancy between simulated deformations and ground-truth data, such as multi-view videos or particle trajectories under known forces. While effective, these methods often require high-quality, dense supervision that is difficult to acquire in real-world settings.

To alleviate this dependency, approaches [17, 25, 54] propose leveraging pre-trained video diffusion models to guide physics discovery via motion distillation losses. Notable examples include PhysDreamer [54] and DreamPhysics [17], which optimize material fields by ensuring simulated dynamics remain consistent with realistic motion priors. Despite their ability to discover physics from sparse inputs, these optimization-based paradigms suffer from extensive computational overhead, often requiring hours of optimization to resolve hundreds of thousands of material points. Furthermore, these methods are characterized by heavy per-scene memorization; for each new scenario, the costly optimization must be executed from scratch, failing to generalize across diverse scenes.

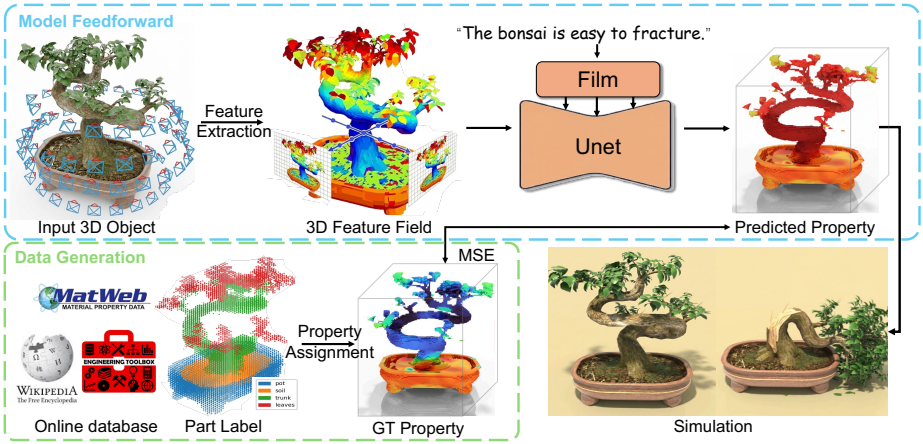


Fig. 2: Overview of our pipeline. Given a 3D object, our framework first decomposes it into semantically distinct components via multi-view feature rendering. Then, a text-conditioned neural network resolves visual ambiguity through cross-modal modulation, predicting volumetric physical properties in a single forward pass. The predicted properties are formulated as a voxel field, ensuring compatibility with diverse 3D representations, *e.g.*, meshes, point clouds, and Gaussian Splatting, for physics simulation.

2.3 Feedforward Physics Property Prediction

Feedforward methods such as Pixie [22] and VoMP [13] have recently emerged to bypass the computational burdens of test-time optimization by directly regressing physical attributes in a single pass. However, these methods rely exclusively on visual cues, which inherently lead to physical ambiguity, *i.e.*, visually similar objects, such as fresh fruit and its plastic replica, can possess drastically different material properties. To resolve this limitation, **AutoPhyX** introduces textual constraints as a crucial semantic prior to resolve the visual ambiguities, enabling the differentiation of materials with similar appearances but distinct physical behaviors. Unlike previous models, our goal is to ensure that the predicted parameters are efficient to compute, controllable, and physically plausible, providing a versatile solution for augmenting diverse 3D representations, including meshes, point clouds, and Gaussian Splatting.

3 Method

Given a 3D object represented with multi-view images and a text prompt, our goal is to predict a dense, physics-ready volumetric field of material parameters, specifically Young's modulus E, Poisson's ratio ν, and density ρ. To achieve this, we propose **AutoPhyX** that operates in two primary stages. First, it extracts 2D OpenCLIP features from multi-view images and lifts them into a dense 3D OpenCLIP voxel grid, explicitly handling self-occlusion via volume rendering and filling the unobservable interior regions to ensure a solid physical volume

(Sec. 3.1). Second, we fuse these 3D visual features with textual embeddings to resolve inherent visual ambiguities, directly decoding this cross-modal representation into a unified voxel field of precise physical parameters ready for downstream simulation (Sec. 3.2). We additionally generate a large-scale dataset to facilitate training and testing, as detailed in Sec. 3.3.

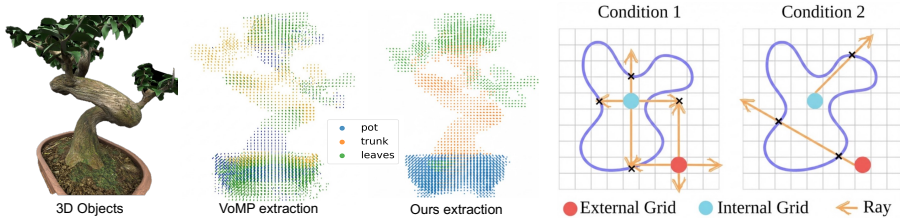
3.1 3D Feature Extraction from 2D Images

Recent methods like VoMP [13] lift 2D features to 3D via naive multi-view averaging. However, ignoring visibility causes severe feature contamination, such as foreground leaves erroneously projecting features onto an occluded trunk, as shown in Figure 3. To address these geometric ambiguities, we shift from heuristic projection to a volumetric rendering framework. We treat the 2D OpenCLIP [18] features as "feature radiance" and optimize a continuous 3D feature field $\Phi : \mathbb{R}^3 \to \mathbb{R}^D$ via end-to-end differentiable rendering. Using a fixed density field σ from a pre-trained RGB Neural Radiance Fields (NeRF), we render the expected 2D feature $\hat{F}(\mathbf{r})$ along a camera ray $\mathbf{r}(t) = \mathbf{o} + t\mathbf{d}$ as $\hat{F}(\mathbf{r}) = \int_{t_n}^{t_f} T(t)\sigma(\mathbf{r}(t))\Phi(\mathbf{r}(t))dt$, where t_n and t_f denote the near and far bounds of the ray, and $T(t) = \exp(-\int_{t_n}^{t} \sigma(\mathbf{r}(s))ds)$ represents the accumulated transmittance. Here, $T(t)$ naturally suppresses occluded regions by dropping to zero behind surfaces, ensuring that the 3D feature field Φ is supervised from viewpoints where the surfaces are truly visible.

While volume rendering captures surface features, physics-based simulations inherently require volumetric properties, such as (E, ν, ρ), defined throughout the object's interior to accurately model stress propagation. To bridge this gap, we fill unobservable internal voids using a hierarchical ray-casting strategy inspired by PhysGaussian [49]. Specifically, following the classic even-odd rule in computer graphics, we identify internal voxels by casting multiple rays to infinity, where a consistently odd number of surface intersections across all rays indicates the voxel is inside, robustly handling complex concavities. Finally, these voxels inherit semantic features via nearest-neighbor interpolation from the surface, yielding a physically and semantically consistent volumetric representation.

3.2 Text-conditioned Physical Parameter Prediction

While vision-based 3D representations, such as 3D CLIP feature field [41], have proven highly effective for open-vocabulary localization and commonsense reasoning, relying solely on pure vision is fundamentally insufficient for predicting physical properties. This limitation arises from the inherent ambiguity of visual appearance. For instance, a rubber toy might look identical whether it is extremely stiff or highly deformable. Therefore, it is essential to introduce textual descriptions as a disambiguating condition, fusing the semantic priors of language with the spatial awareness of 3D visual features to guide the physical reasoning process.



(a) 3D Feature Extraction Comparison **(b) Internal Voxel Filling**

Fig. 3: Feature extraction and volumetric filling. (a) Unlike direct averaging (e.g., VoMP [13]), which suffers from feature contamination by ignoring visibility, our NeRF-based extraction preserves 3D structural integrity and spatial consistency. **(b)** Hierarchical classification for internal voxel filling. Condition 1 performs coarse enclosure detection via six-direction boundary checking. Condition 2 provides topological refinement using ray-surface intersection counts to filter candidates and achieve accurate filling in complex manifolds.

Specifically, the textual description is first processed by a frozen OpenCLIP text encoder to extract a dense semantic embedding, denoted as e_{text}. Note that since both the 3D visual voxels and the textual embeddings are derived from the OpenCLIP latent space, they are inherently aligned, which avoids the need for complex cross-modal projection or pre-processing modules. To achieve the cross-modal fusion, we draw inspiration from Feature-wise Linear Modulation (FiLM) [38], which demonstrates remarkable success in visual reasoning tasks. Rather than naively concatenating this embedding with visual features, e_{text} acts as a direct and highly effective global condition to modulate the 3D OpenCLIP visual voxels at multiple scales.

In our design, e_{text} is passed through dedicated Multi-Layer Perceptrons (MLP) to predict layer-specific modulation parameters: the scaling factor γ and the shifting factor β. Let F_i represent the intermediate 3D voxel feature map at the i-th layer of our U-Net. The FiLM layer applies a channel-wise affine transformation to modulate this feature map:

$$F_i' = \gamma_i(e_{text}) \cdot F_i + \beta_i(e_{text}). \tag{1}$$

By interleaving these FiLM layers throughout the 3D U-Net, the network learns to dynamically emphasize or suppress specific visual feature channels based on linguistic cues. This multi-scale, text-driven modulation ensures that the final extracted features are highly attuned to the physical attributes specified in the text, thereby enabling the accurate regression of the target physical parameters. We present a detailed comparison between our design and other multi-modal fusion modules in the *Supplementary Material*.

The physics loss $\mathcal{L}_{phys}$ is computed by enforcing supervision exclusively on the occupied voxels $\mathcal{G}$. Let $\mathcal{K} = \{E, \nu, \rho\}$ denote the set of target physical properties, corresponding to Young's modulus, Poisson's ratio, and density, respectively.

We formulate the total loss as:

$$\mathcal{L}_{phys} = \frac{1}{N_{occ}} \sum_{p \in \mathcal{G}} \sum_{k \in \mathcal{K}} \|\hat{k}(p) - k^{GT}(p)\|^2, \tag{2}$$

where N_{occ} represents the total count of valid voxels, and $\hat{k}(p)$, $k^{GT}(p)$ denote the predicted and ground-truth physical parameters at voxel p.

3.3 Dataset Generation

Unlike Pixie [22] and VoMP [13], which supervise the prediction of each object by a single deterministic set of physical parameters, our approach resolves the inherent ambiguity of prediction by introducing text description as conditioning signals. This formulation supports downstream applications that require flexible object physics, *e.g.*, increasing the robustness of robot manipulation policy by varying the potential plausible physics for the same object during training. However, such text-conditioned training data are currently unavailable at scale, motivating our automated pipeline for large-scale data generation.

Semantic Part Segmentation. To achieve precise text-driven control, *e.g.*, accurately specifying whether a tree trunk is hardwood or softwood, or adjusting the stiffness of a car's tires, it is required that our dataset provide accurate part-level segmentation. To this end, we first extract 3D OpenCLIP voxels for each 3D object as in Sec. 3.1. OpenCLIP voxels have been successfully and widely adopted in object grounding and reasoning [41], and their text-aligned semantic features enable segmentation of objects based on corresponding text keywords.

To mitigate the ambiguity associated with single-view observations, we uniformly render 15 images from the upper hemisphere of each object. We then prompt VLM [12] to generate a text list encompassing all constituent parts it identifies for the object. Upon obtaining these text keywords, we compute the segmentation on the OpenCLIP voxels based on the cosine similarity between the text-visual features. To ensure robustness, we repeatedly generate five distinct sets of keywords per object and render their corresponding 3D segmentations, as illustrated in Figure 3a. The VLM finally selects the highest-quality segmentation from these five candidates, which effectively prevents incorrect segmentations caused by the use of uncommon vocabulary for part names.

Text-Guided Property Assignment. After part-level segmentation, the next step is to generate the physical properties paired with text descriptions. To resolve the physical ambiguity from pure visual inputs, our guideline is to generate text first to guide the property assignment. For each object, we randomly select a subset of parts and prompt VLM [12] to generate physically diverse but plausible textual descriptions, which act as the guideline for parameter assignment. To ground these assignments and prevent potential VLM hallucination, for each description, we prompt VLM to query the corresponding physics parameter range from online databases, *i.e.*, Matweb [30], Wikipedia [45], and The Engineering Toolbox [15]. Once the physics ranges are acquired, we leverage

VLM to sample a corresponding parameter in the valid range. For instance, the description "This is mahogany that might sink to the bottom," would lead to a density value near the upper end of the wood's density range (800 kg/m^3 to 1300 kg/m^3), whereas "This is softwood that can float on water" yields a lower density. When an object comprises multiple parts, any unmentioned parts are assigned the mean value of their valid physical ranges. In addition, to ensure robust physical parameter assignment in the absence of detailed textual guidance, for each object, we include a default text description "This is a [object name]," where all parts are assigned the mean value in their valid ranges.

3D Asset Collection To build a diverse and comprehensive dataset, we acquire our 3D assets from two distinct sources. First, we collect approximately 1,000 high-quality 3D meshes from the Objaverse dataset. Second, to capture the complexity of real-world objects, we gather diverse 2D images via Google Search and utilize SAM 3D Objects [44] to reconstruct approximately 700 3D Gaussian Splatting (GS) models. This yields a total base of 1,700 unique 3D assets. As detailed above, we generate 8 distinct textual descriptions and their corresponding physical parameters for each object. Consequently, our dataset, **Text2PhyX**, contains $1700 \times 8 = 13600$ data triplets, with each triplet comprising a 3D object representation, a text description, and the ground-truth physical parameters. We partition this collection into 12,600 training samples and 1,000 testing samples. A gallery of **Text2PhyX** is included in *Supplementary Material*.

4 Experiment

We conduct a comprehensive evaluation of **AutoPhyX** to validate its ability to recover accurate physical constants from visual and linguistic inputs. Our experiments focus on two primary objectives: (1) **Quantitative Analysis**, where we measure the precision of predicted physics properties on our dataset **Text2PhyX** and a real-world dataset ABO-500; and (2) **Qualitative Simulation**, where we demonstrate the practical utility of our framework by applying the predicted physics properties to realistic physics-based simulations, containing both Gaussian Splatting and mesh. Through these evaluations, we show that **AutoPhyX** effectively bridges the gap between high-level semantic descriptions and low-level physical dynamics in an end-to-end manner.

Dataset and Settings. We evaluate text-conditioned physical parameter prediction on the test set of **Text2PhyX**, detailed in Sec. 3.3. We measure the discrepancy between the estimated and ground-truth physical parameters. Furthermore, we assess the visual and dynamic quality of renderings from simulations driven by the predicted properties and the ground-truth.

Furthermore, to rigorously assess real-world generalization, we incorporate the ABO-500 dataset as an additional out-of-distribution test set. The ABO-500 dataset contains the density and mass of 500 real-world objects, which allows us to evaluate the discrepancy between our predicted physical properties and the real-world ground truth. Since this dataset lacks detailed physical descriptions, we solely use text prompts following the template: "This is a [object name]."

Table 1: Quantitative comparison on Text2PhyX. We evaluate both rendering quality (PSNR, SSIM, LPIPS) and physical property prediction accuracy (Avg. Phys. err, $\log E$ err, ν err, $\log \rho$ err). "–" indicates that the method cannot predict the corresponding physical parameter. **Bold** and underline denote the best and second-best results, respectively. Our method, particularly when equipped with OpenCLIP, significantly outperforms all baselines across both rendering and physics metrics.

Method	PSNR ↑	SSIM ↑	LPIPS ↓	Avg. Phys. err ↓	$\log E$ err ↓	ν err ↓	$\log \rho$ err ↓
DreamPhysics							
1 epoch	19.412	0.881	0.212	–	2.351	–	–
25 epochs	19.515	0.882	0.185	–	1.405	–	–
50 epochs	19.223	0.881	0.178	–	1.362	–	–
OmniPhysGS							
1 epoch	17.955	0.883	0.325	–	–	–	–
2 epochs	17.912	0.883	0.310	–	–	–	–
5 epochs	17.880	0.884	0.305	–	–	–	–
NeRF2Physics	18.554	0.889	0.245	0.841	0.585	0.455	0.982
Gemini (avg.)	21.124	0.869	0.205	0.218	0.227	0.235	0.192
Pixie (uncond. eval only)							
Occupancy	17.901	0.868	0.285	0.122	0.145	0.121	0.101
RGB	18.695	0.863	0.262	0.121	0.191	0.075	0.097
Clip	23.421	0.908	0.092	0.078	0.062	0.065	0.108
Ours (w/ Clip)	25.105	0.916	0.091	0.069	0.058	0.061	0.089
Ours (w/ OpenClip)	**26.012**	**0.921**	**0.089**	**0.059**	**0.042**	**0.051**	**0.083**

Implementation Detail. **AutoPhyX** is trained using the Adam optimizer with the initial learning rate 10^{-4} on 2 NVIDIA RTX A800 GPUs. Training takes 2 days with a batch size of 4 per GPU and 100 epochs. For both training and metric computation, we apply a log transform to E and ρ, and normalize all $\log E$, ν, and $\log \rho$ values to the range $[-1, 1]$. For detailed model architecture and the choice of the OpenCLIP version, please refer to the *Supplementary Material*.

Baselines. We compare **AutoPhyX** against several representative methods. Both OmniPhysGS [25] and DreamPhysics [17] utilize video diffusion models to optimize the physics simulation process online. NeRF2Physics [51] uses GPT [1] to generate material text prompts and their physical properties, mapping them to 3D points via text-image CLIP similarity. We also test representative VLM, Gemini, which we ask to predict the part name and corresponding physics property, and then assign them back to the object corresponding to OpenCLIP feature similarity. To alleviate the randomness of VLM output, we conduct 5 queries and take the average as the final result of Gemini. Pixie [22] is a recent feed-forward physics generation method that relies solely on visual features. Since the original dataset for Pixie is not publicly available, we retrain it on our dataset to ensure a fair comparison. Furthermore, as Pixie is the only method in Tab. 1 that does not support textual descriptions as input, we evaluate it exclusively

under the default unconditional setting described in Sec. 3.3. Finally, we exclude VoMP [13] from our evaluation because its source code is not publicly available.

Evaluation Metrics. In **Text2PhyX**, we evaluate using physical and visual metrics. For physical accuracy, we compute the Mean Squared Error (MSE) for Young's modulus E, Poisson's ratio ν, and density ρ. To handle large magnitude variations, E and ρ are evaluated in log-space ($\log_{10}$). For visual fidelity, we simulate and render images with the predicted properties using PhysGaussian [49] for Gaussian Splatting, and Isaac Sim [29] for meshes. The rendered frames are then evaluated against ground truth using Peak Signal-to-Noise Ratio (PSNR), Structural Similarity Index Measure (SSIM), and Learned Perceptual Image Patch Similarity (LPIPS).

On the real-world ABO-500 mass dataset, we report four metrics: ADE (Average Displacement Error), ARE (Average Relative Error), ALDE (Average Log Displacement Error), and MnRE (Minimum Ratio Error). Specifically, ADE measures the absolute numerical difference in weight, while ARE captures the percentage of this error relative to the object's true weight, which is crucial for lighter items. Furthermore, ALDE evaluates the error in logarithmic space. MnRE avoids bias toward systematic over- or underestimation and reduces sensitivity to outliers, making it particularly effective for evaluating physical quantity predictions across heterogeneous samples.

Quantitative Result As shown in Tab. 1, our method exhibits a significant performance margin over existing baselines. **AutoPhyX** with the OpenCLIP features achieves a PSNR of 26.012, outperforming the state-of-the-art Pixie by approximately 2.6 dB. This significant improvement is primarily attributed to our cross-modal modulation utilizing the text conditions, coupled with OpenCLIP's richer semantic representations compared to standard CLIP; they together empower our model to make finer-grained physical predictions. Pixie, while making reasonable predictions, is constrained to the default physical conditions because it solely takes CLIP voxel features as input. This reliance on purely visual input suffers from visual ambiguity, which is more evident on the ABO-500 object mass estimation benchmark. As shown in Tab. 2, our method consistently improves accuracy over NeRF2Physics and Pixie, achieving lower ALDE, ADE, ARE, and higher MnRE, which suggests that **AutoPhyX** not only achieves a good performance on text-conditioned prediction on our dataset, but also generalizes well in the out-of-distribution real-world dataset across diverse physical scenarios.

OmniPhysGS and DreamPhysics both achieve low PSNR, SSIM, and LPIPS. These poor results stem from the fact that OmniPhysGS exclusively optimizes the constitutive model, whereas DreamPhysics focuses solely on Young's modulus. In reality, real-world dynamics are governed by a coupled set of physical properties. Because these methods isolate and optimize only a single physical dimension, their overall degrees of freedom of simulation are severely limited; this is another drawback for the per-scene optimization methods, as the joint optimization of multiple parameters is substantially more challenging. Since no

Table 2: Quantitative comparison on ABO-500 dataset. ALDE: average log displacement error; ADE: average displacement error; ARE: average relative error; MnRE: minimum ratio error $\left(\frac{1}{N}\sum_{i=1}^{N}\min\left(\frac{y_i}{\hat{y}_i}, \frac{\hat{y}_i}{y_i}\right)\right)$. Lower is better for ALDE/ADE/ARE; higher is better for MnRE.

Method	ALDE (↓)	ADE (↓)	ARE (↓)	MnRE (↑)
NeRF2Physics	0.732	11.826	0.989	0.573
Pixie	0.782	12.565	0.971	0.568
Ours	**0.652**	**9.433**	**0.901**	**0.582**

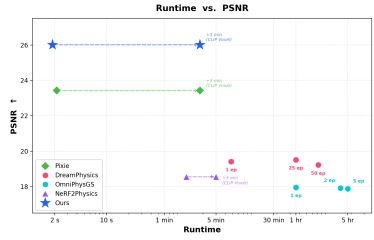


Fig. 4: Runtime vs. PSNR. Ours achieves the best PSNR (26.01) in under 2s of inference, orders of magnitude faster than optimization-based methods.

single physical parameter can comprehensively capture the complex, non-linear dynamics required for accurate 3D simulation, the resulting simulated results deviate markedly from the ground-truth visual observations. In contrast, benefiting from large-scale training, our method unlocks richer and more controllable simulations by simultaneously predicting three physical properties, thereby substantially outperforming both approaches across all evaluation metrics.

Since NeRF2Physics directly relies on GPT for physical property predictions, it inherits inaccurate physical priors and VLM hallucinations. As a result, its performance in terms of physical accuracy and visual fidelity is relatively low. Similar to NeRF2Physics, directly leveraging Gemini as an agent that predicts physics properties cannot achieve high precision. While **Text2PhyX** also employs Gemini for part segmentation and physical property retrieval during data generation, we mitigate these limitations through a multiple-sampling strategy for robust part segmentation and text-guided property assignment. Furthermore, we ground our queries in authoritative online databases rather than relying solely on the VLM's internal knowledge for physical properties. This also highlights a key *asymmetry*: VLMs can generate coherent part descriptions and then produce compatible physical parameters in a forward manner, but they struggle with the inverse problem that requires specific property estimation from a given text description, because their internal physics knowledge is not calibrated for accurate, instance-specific parameter estimation. Consequently, although we utilize Gemini for data generation, our method leverages large-scale training to improve generalizability and substantially outperform pure VLM-based baselines across all evaluation metrics.

Runtime. Figure 4 plots PSNR versus runtime (log-scale). Our method attains the highest PSNR (26.01) with 2s inference, while optimization-based baselines (DreamPhysics and OmniPhysGS) require minutes to hours, depending on the number of epochs, and yield lower PSNR. For Pixie, NeRF2Physics, and our method, we additionally show a second point that accounts for approximately 3 minutes of CLIP voxel feature extraction as input preprocessing.

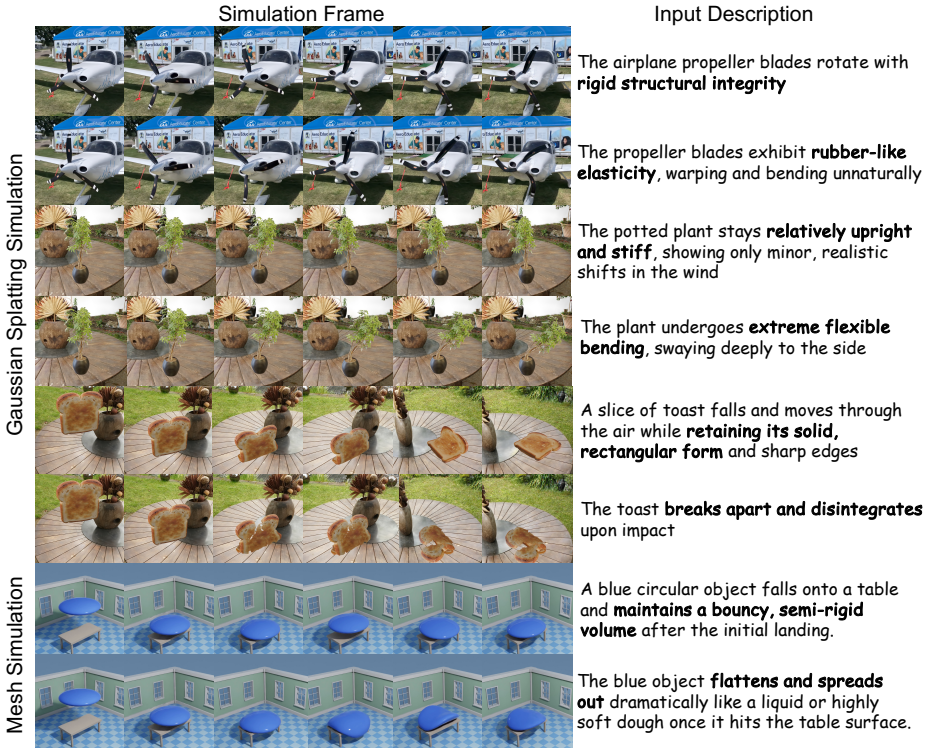


Fig. 5: Qualitative verification of physical parameter prediction. To evaluate the effectiveness of our parameter estimation, we visualize simulations under identical boundary conditions but with different predicted physical properties. Each pair of rows demonstrates how various descriptions alter the simulation behavior: Rows 1–2 for rigid vs. elastic propeller blades, Rows 3–4 for stiff vs. flexible plant stems, Rows 5–6 for structural integrity vs. brittle fracture of toast, and Rows 7–8 for semi-rigid vs. large deformation of a soft body.

Qualitative Results As illustrated in Fig. 5, the versatility of our voxelized prediction approach allows the predicted physical properties to be integrated into diverse simulation pipelines. For the first six rows, we employ the MPM to drive the dynamic behavior of Gaussian Splatting representations, capturing complex physical effects under varying material properties: from rigid rotation to elastic warping of the propeller (Rows 1-2), from gentle swaying to pronounced bending of the ficus in the wind (Rows 3-4), and from an intact collision to brittle fracture of the falling bread (Rows 5-6).

Furthermore, to demonstrate the cross-representation capability, the final two rows present deformable object simulations conducted in Isaac Sim using the FEM. Despite the change in both the physical solver and the 3D representation, our model consistently predicts accurate parameters that govern realistic material behaviors, such as volume-preserving bouncing versus large deforma-

tion. This highlights that our method learns intrinsic physical attributes that are independent of specific simulators or 3D representations.

As illustrated in Fig. 6, our voxel-based prediction framework achieves significantly higher precision in material decomposition compared to existing methods. A key limitation of NeRF2Physics is its lack of structural granularity, which predicts nearly identical physical attributes for different parts, such as assigning uniform Young's modulus values to both the foliage and the trunk. On the other hand, Pixie struggles with material boundaries at the base of the object, predicting that the soil and the ceramic pot have the same physical properties. In contrast, our method successfully recovers the sharp discontinuities between distinct materials. It accurately distinguishes the rigid trunk from the soft leaf clusters and maintains a clear boundary between the soil and the rigid container. This high-fidelity voxelized heatmap demonstrates that our model captures the intrinsic semantic-physical correlations of the scene rather than just global averages.

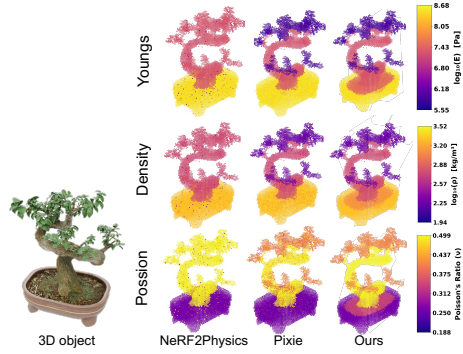


Fig. 6: Voxel prediction across different methods. Compared to baselines that struggle with structural granularity (NeRF2Physics) or material boundaries (Pixie), our framework achieves highly precise material decomposition.

5 Conclusion and Future Work

We presented **AutoPhyX**, a novel text-conditioned feed-forward framework that automates the assignment of dense physical properties for diverse 3D assets. Addressing the critical bottleneck of manual parameter tuning and the inherent material ambiguities of pure vision approaches, our method leverages a unified voxel field representation to seamlessly bridge high-level linguistic semantics with low-level physical dynamics. To support this cross-modal paradigm and overcome the critical scarcity of training data, we designed an automated, VLM-driven generation pipeline and constructed a large-scale, multi-part 3D physics dataset **Text2PhyX**.

Extensive experiments validate the superiority of **AutoPhyX**. Quantitatively, it achieves state-of-the-art accuracy on both our dataset and the real-world ABO-500 benchmark. Qualitatively, our method seamlessly integrates into diverse simulation pipelines, including MPM and FEM, across distinct objects such as a bonsai, a propeller, and bread. These results demonstrate its immense potential for scaling to larger, complex interactive environments. In the future, we plan to extend our framework by incorporating additional physical properties, such as surface friction and damping coefficients.

References

1. Achiam, J., Adler, S., Agarwal, S., Ahmad, L., Akkaya, I., Aleman, F.L., Almeida, D., Altenschmidt, J., Altman, S., Anadkat, S., et al.: Gpt-4 technical report. arXiv preprint arXiv:2303.08774 (2023) 10
2. Baraff, D.: An introduction to physically based modeling: rigid body simulation i—unconstrained rigid body dynamics. SIGGRAPH course notes **82** (1997) 1
3. Bardenhagen, S.G., Kober, E.M., et al.: The generalized interpolation material point method. Computer Modeling in Engineering and Sciences (2004) 1
4. Bathe, K.J.: Finite element method. Wiley encyclopedia of computer science and engineering (2007) 1
5. Bathe, K.J., Ramm, E., Wilson, E.L.: Finite element formulations for large deformation dynamic analysis. International journal for numerical methods in engineering (1975) 1
6. Belytschko, T., Liu, W.K., Moran, B., Elkhodary, K.: Nonlinear finite elements for continua and structures (2014) 1
7. Cao, Z., Chen, Z., Pan, L., Liu, Z.: Physx-3d: Physical-grounded 3d asset generation (2025)
8. Chen, B., Jiang, H., Liu, S., Gupta, S., Li, Y., Zhao, H., Wang, S.: Physgen3d: Crafting a miniature interactive world from a single image. In: Conference on Computer Vision and Pattern Recognition (CVPR) (2025) 4
9. Chen, L.Q.: Phase-field models for microstructure evolution. Annual review of materials research (2002) 1
10. Chen, Y., Hu, Y., Sun, L., Kusnur, T., Herlant, L., Jiang, C.: Empm: Embodied mpm for modeling and simulation of deformable objects. arXiv preprint arXiv:2601.17251 (2026) 1
11. Collins, J., Goel, S., Deng, K., Luthra, A., Xu, L., Gundogdu, E., Zhang, X., Vicente, T.F.Y., Dideriksen, T., Arora, H., 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. pp. 21126–21136 (2022) 3
12. Comanici, G., Bieber, E., Schaekermann, M., Pasupat, I., Sachdeva, N., Dhillon, I., Blistein, M., Ram, O., Zhang, D., Rosen, E., et al.: Gemini 2.5: Pushing the frontier with advanced reasoning, multimodality, long context, and next generation agentic capabilities. arXiv preprint arXiv:2507.06261 (2025) 8
13. Dagli, R., Xiang, D., Modi, V., Loop, C., Tsang, C.F., Chen, A.H., Hu, A., State, G., Levin, D.I.W., Shugrina, M.: Vomp: Predicting volumetric mechanical property fields (2025), `https://arxiv.org/abs/2510.22975` 2, 5, 6, 7, 8, 11
14. De Vaucorbeil, A., Nguyen, V.P., Sinaie, S., Wu, J.Y.: Material point method after 25 years: Theory, implementation, and applications. Advances in applied mechanics (2020) 1, 3
15. Engineering ToolBox: The engineering toolbox. `https://www.engineeringtoolbox.com/` 8
16. Hsu, H.Y., Lin, C.H., Zhai, A.J., Xia, H., Wang, S.: Autovfx: Physically realistic video editing from natural language instructions. In: International Conference on 3D Vision (3DV) (2025) 4
17. Huang, T., Zhang, H., Zeng, Y., Zhang, Z., Li, H., Zuo, W., Lau, R.W.: Dreamphysics: Learning physics-based 3d dynamics with video diffusion priors. In: Proceedings of the AAAI Conference on Artificial Intelligence. vol. 39, pp. 3733–3741 (2025) 2, 4, 10

18. Ilharco, G., Wortsman, M., Wightman, R., Gordon, C., Carlini, N., Taori, R., Dave, A., Shankar, V., Namkoong, H., Miller, J., Hajishirzi, H., Farhadi, A., Schmidt, L.: Openclip (Jul 2021). https://doi.org/10.5281/zenodo.5143773, https://doi.org/10.5281/zenodo.5143773, if you use this software, please cite it as below. 3, 6
19. Jiang, C., Schroeder, C., Teran, J., Stomakhin, A., Selle, A.: The material point method for simulating continuum materials. In: Acm siggraph 2016 courses, pp. 1–52 (2016) 1, 3
20. Jiang, H., Hsu, H.Y., Zhang, K., Yu, H.N., Wang, S., Li, Y.: Phystwin: Physics-informed reconstruction and simulation of deformable objects from videos. In: Proceedings of the IEEE/CVF International Conference on Computer Vision. pp. 7219–7230 (2025) 4
21. Joshuah, W., Yunuo, C., Minchen, L., Yu, F., Ziyin, Q., Jiecong, L., Meggie, C., Chenfanfu, J.: Anisompm: Animating anisotropic damage mechanics. ACM Trans. Graph. **39**(4) (2020) 1
22. Le, L., Lucas, R., Wang, C., Chen, C., Jayaraman, D., Eaton, E., Liu, L.: Pixie: Fast and generalizable supervised learning of 3d physics from pixels (2025), https://arxiv.org/abs/2508.17437 2, 5, 8, 10
23. Li, X., Qiao, Y.L., Chen, P.Y., Jatavallabhula, K.M., Lin, M., Jiang, C., Gan, C.: Pac-nerf: Physics augmented continuum neural radiance fields for geometry-agnostic system identification. arXiv preprint arXiv:2303.05512 (2023) 4
24. Li, Z., Yu, H.X., Liu, W., Yang, Y., Herrmann, C., Wetzstein, G., Wu, J.: Wonderplay: Dynamic 3d scene generation from a single image and actions. In: Proceedings of the IEEE/CVF International Conference on Computer Vision. pp. 9080–9090 (2025) 4
25. Lin, Y., Lin, C., Xu, J., Mu, Y.: Omniphysgs: 3d constitutive gaussians for general physics-based dynamics generation. arXiv preprint arXiv:2501.18982 (2025) 2, 4, 10
26. Liu, M., Liu, G.R.: Restoring particle consistency in smoothed particle hydrodynamics. Applied numerical mathematics **56**(1), 19–36 (2006) 1
27. Luo, Z., Cui, Z., Luo, S., Chu, M., Li, M.: Vr-doh: Hands-on 3d modeling in virtual reality. ACM Transactions on Graphics (TOG) **44**(4), 1–12 (2025) 1
28. Ma, Y., Li, Y., Ye, J., Wu, Z., Zhang, W., Gao, L., Jin, Z.: Fastphysgs: Accelerating physics-based dynamic 3dgs simulation via interior completion and adaptive optimization. arXiv preprint arXiv:2602.01723 (2026) 4
29. Makoviychuk, V., Wawrzyniak, L., Guo, Y., Lu, M., Storey, K., Macklin, M., Hoeller, D., Rudin, N., Allshire, A., Handa, A., State, G.: Isaac gym: High performance gpu-based physics simulation for robot learning (2021), https://arxiv.org/abs/2108.10470 11
30. MatWeb, LLC: Online materials information resource - matweb. https://www.matweb.com/ 8
31. Moës, N., Dolbow, J., Belytschko, T.: A finite element method for crack growth without remeshing. International journal for numerical methods in engineering (1999) 1
32. Monaghan, J.J.: Smoothed particle hydrodynamics. In: Annual review of astronomy and astrophysics. Vol. 30 (A93-25826 09-90), p. 543-574. (1992) 1
33. Monaghan, J.J.: Smoothed particle hydrodynamics. Reports on progress in physics (2005) 1
34. Monaghan, J.J.: Smoothed particle hydrodynamics and its diverse applications. Annual Review of Fluid Mechanics (2012) 1

35. Morris, J.P.: Analysis of smoothed particle hydrodynamics with applications. Monash University Australia (1996) 1
36. Mou, L., Chen, J.K., Wang, Y.X.: Instruct 4d-to-4d: Editing 4d scenes as pseudo-3d scenes using 2d diffusion. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 20176–20185 (2024) 1
37. Murthy, J.K., Macklin, M., Golemo, F., Voleti, V., Petrini, L., Weiss, M., Considine, B., Parent-Lévesque, J., Xie, K., Erleben, K., et al.: gradsim: Differentiable simulation for system identification and visuomotor control. In: International conference on learning representations (2020) 4
38. Perez, E., Strub, F., de Vries, H., Dumoulin, V., Courville, A.C.: Film: Visual reasoning with a general conditioning layer. In: AAAI (2018) 7
39. Randles, P.W., Libersky, L.D.: Smoothed particle hydrodynamics: some recent improvements and applications. Computer methods in applied mechanics and engineering **139**(1-4), 375–408 (1996) 1
40. Shao, R., Zheng, Z., Tu, H., Liu, B., Zhang, H., Liu, Y.: Tensor4d: Efficient neural 4d decomposition for high-fidelity dynamic reconstruction and rendering. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 16632–16642 (2023) 1
41. Shen, W., Yang, G., Yu, A., Wong, J., Kaelbling, L.P., Isola, P.: Distilled feature fields enable few-shot language-guided manipulation. In: 7th Annual Conference on Robot Learning (2023), https://openreview.net/forum?id=RbOnGIt_kh5 6, 8
42. Sołowski, W., Sloan, S.: Evaluation of material point method for use in geotechnics. International Journal for Numerical and Analytical Methods in Geomechanics (2015) 1
43. Swegle, J.W., Hicks, D.L., Attaway, S.W.: Smoothed particle hydrodynamics stability analysis. Journal of computational physics **116**(1), 123–134 (1995) 1
44. Team, S.D., Chen, X., Chu, F.J., Gleize, P., Liang, K.J., Sax, A., Tang, H., Wang, W., Guo, M., Hardin, T., Li, X., Lin, A., Liu, J., Ma, Z., Sagar, A., Song, B., Wang, X., Yang, J., Zhang, B., Dollár, P., Gkioxari, G., Feiszli, M., Malik, J.: Sam 3d: 3dfy anything in images (2025), https://arxiv.org/abs/2511.16624 9
45. Wikipedia contributors: Wikipedia, the free encyclopedia. https://www.wikipedia.org/, accessed: 2026-03-05 8
46. Wolper, J., Fang, Y., Li, M., Lu, J., Gao, M., Jiang, C.: Cd-mpm: continuum damage material point methods for dynamic fracture animation. ACM Transactions on Graphics (TOG) (2019) 1
47. Wolper, J., Gao, M., Lüthi, M.P., Heller, V., Vieli, A., Jiang, C., Gaume, J.: A glacier–ocean interaction model for tsunami genesis due to iceberg calving. Communications Earth & Environment **2**(1), 130 (2021) 1
48. Wu, G., Yi, T., Fang, J., Xie, L., Zhang, X., Wei, W., Liu, W., Tian, Q., Wang, X.: 4d gaussian splatting for real-time dynamic scene rendering. In: Conference on Computer Vision and Pattern Recognition (CVPR) (2024) 1
49. Xie, T., Zong, Z., Qiu, Y., Li, X., Feng, Y., Yang, Y., Jiang, C.: Physgaussian: Physics-integrated 3d gaussians for generative dynamics. In: Conference on Computer Vision and Pattern Recognition (CVPR) (2024) 6, 11
50. Xiong, X., Hu, C., Lin, C., Ma, P., Gan, C., Du, T.: Topogaussian: Inferring internal topology structures from visual clues. arXiv preprint arXiv:2503.12343 (2025) 4
51. Zhai, A.J., Shen, Y., Chen, E.Y., Wang, G.X., Wang, X., Wang, S., Guan, K., Wang, S.: Physical property understanding from language-embedded feature fields. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 28296–28305 (2024) 2, 3, 4, 10

52. Zhan, J., Li, Z., Yu, H.X., Wu, J.: Perpetualwonder: Long-horizon action-conditioned 4d scene generation. arXiv preprint arXiv:2602.04876 (2026) 4
53. Zhang, K., Li, B., Hauser, K., Li, Y.: Particle-grid neural dynamics for learning deformable object models from rgb-d videos. arXiv preprint arXiv:2506.15680 (2025) 4
54. Zhang, T., Yu, H.X., Wu, R., Feng, B.Y., Zheng, C., Snavely, N., Wu, J., Freeman, W.T.: Physdreamer: Physics-based interaction with 3d objects via video generation. In: European Conference on Computer Vision (ECCV) (2024) 2, 4
55. Zhong, L., Yu, H.X., Wu, J., Li, Y.: Reconstruction and simulation of elastic objects with spring-mass 3d gaussians. In: European Conference on Computer Vision. pp. 407–423. Springer (2024) 4