Supplementary Material

A. 3D Open-vocabulary Segmentation

2D Open-vocabulary Segmentation. Inspired by prior successful works [7], we innovatively introduce the integration of 2D open-vocabulary detector models, such as Grounding DINO [6], promptable 2D segmentation models, such as SAM [4], image tagging models like RAM [12]. The integrated 2D open-vocabulary model can automatically segment objects within images without the need for any textual input.

Specifically, given an input image, we first employ an image tagging model, RAM [12] to get the tags of the image. Then, given the tags, we employ Grounding DINO [6] to generate precise boxes for objects or regions within the image by leveraging the textual information in tags as condition. Subsequently, the annotated boxes obtained through Grounding DINO serve as the box prompts for SAM to generate precise mask annotations. By leveraging the capabilities of these robust expert models, our method enables the automatic labeling of an entire image.

3D Open-vocabulary Segmentation. After 2D open-vocabulary segmentation, the segmented images contain rich semantic features for every object in the 3D scene. We effectively lift these 2D masks to segment anything in the 3D scene via radiance fields rendering.

Given a pre-trained 3D scene, inspired by recent works [10, 13], we preserve all attributes of the Gaussians, but add a semantic attribute to integrate semantic information for each Gaussian. Then, to assign each 2D mask a unique ID in the 3D scene, we need to associate the masks of the same identity across different views. We employ a well-trained zero-shot tracker [1] to propagate and associate these masks.

In addition to the existing Gaussian properties, we introduce a new parameter, semantic attribute, to each Gaussian. The semantic attribute is a learnable and compact vector, which is used to distinguish semantic categories in whole 3D scene. To optimize the introduced attribute of each Gaussian, we render semantic attribute into 2D images in a differentiable manner as:

$$\mathcal{S} = \sum_{p \in \mathcal{N}} y_p \alpha_p \prod_{j=1}^{p-1} (1 - \alpha_j), \tag{1}$$

where $\mathcal{S}_k$ represents the 2D semantic labels of pixel k, derived from Gaussian point semantic attributes via α-blending. Here, y_p denotes the semantic attribute of the 3D Gaussian point p, and α_p is the influence factor of this point in rendering pixels. After associating 2D instance labels across each training view, we apply the grouping loss [10] and 3D Gaussian reconstruction loss [3] to supervise the optimization progress.

Extracting objects from 3DGS introduces holes, which we inpaint using LaMa [8]. This inpainting ensures more natural results when objects undergo displacement due to external forces.

B. Implementation Details for Baselines

In this section, we elaborate on the implementation details of baselines used for comparison to our proposed method. For PhysDreamer [11], we used the pre-trained models provided in the official code repository[1], as the training code is not made available. For Physics3D [5], we train the models using the code from official code repository[2]. For DreamPhysics [2], we train the models using the code from official code repository[3]. All other hyperparameters remain unchanged. The trained models are then used for qualitative evaluation.

C. User Study

We use Tencent Survey[4] to recruit participants for the human preference evaluation. The survey is fully anonymized. For each scenario, we provided video clips and asked the participants to give each video a score. A total of 41 volunteers participated in the study, including 3 professionals from the 3D art industry.

D. Video Visualization

We provide generated videos in the project page[5].

E. More Analysis about Material Property Distribution Prediction

In our paper, we train an MPDP model using part of the data from Physics3D [5]. However, with the advancement of 3D content creation networks, such as LGM [9], we can generate diverse objects through these methods and utilize Physics3D for m material property distribution prediction to create additional training data. This approach has the potential to further enhance the performance of our model and represents a direction for our future work.

F. More Details about Material Point Method (MPM)

The Material Point Method (MPM) is an advanced numerical technique for simulating the behavior of continuum materials. It discretizes a material body into material points,

[1] https://github.com/a1600012888/PhysDreamer
[2] https://github.com/liuff19/Physics3D
[3] https://github.com/tyhuang0428/DreamPhysics
[4] https://wj.qq.com/index.html
[5] https://sim-gs.github.io/

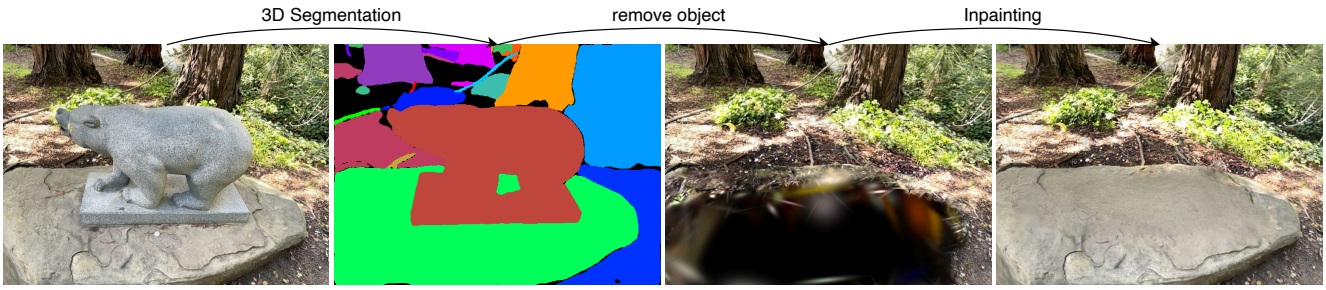

Figure 1. The whole pipeline for 3D Open-vocabulary Segmentation.

commonly referred to as particles, which carry essential properties such as mass, velocity, deformation gradient, and stress. These particles interact with a background computational grid, which facilitates spatial derivative calculations and the application of external forces.

MPM consists of two primary phases: 1) Particle-to-Grid (P2G) Transfer: Particles transfer their properties to the grid, enabling the computation of global quantities such as forces and accelerations. 2) Grid-to-Particle (G2P) Transfer: Updated grid values, such as velocities and positions, are mapped back to the particles, ensuring their motion aligns with the computed dynamics.

This dual transfer mechanism allows MPM to efficiently handle large deformations and complex interactions in continuum materials.

Particle-to-Grid (P2G) Transfer. During this phase, the particles' properties, such as mass and momentum, are mapped to the computational grid using interpolation functions. The mass at a grid node i is computed as:

$$m_i^n = \sum_p w_{ip}^n m_p,$$

where m_p is the mass of particle p, and w_{ip}^n is the interpolation weight (often derived from a B-spline kernel) between particle p and grid node i. The momentum at the grid node is similarly updated:

$$m_i^n \mathbf{v}_i^n = \sum_p w_{ip}^n m_p \left(\mathbf{v}_p^n + \mathbf{C}_p^n(\mathbf{x}_i - \mathbf{x}_p^n)\right),$$

where $\mathbf{v}_p^n$ is the velocity of particle p, $\mathbf{C}_p^n$ represents the affine velocity field gradient, and $\mathbf{x}_i$ and $\mathbf{x}_p^n$ are the positions of the grid node and particle, respectively.

Grid Update. Once particle properties are transferred, grid velocities are updated by accounting for external forces, internal stresses, and gravity. The velocity at grid node i is computed as:

$$\mathbf{v}_i^{n+1} = \mathbf{v}_i^n - \frac{\Delta t}{m_i^n} \sum_p \boldsymbol{\tau}_p^n \nabla w_{ip}^n V_p^0 + \Delta t \mathbf{g},$$

where Δt is the time step, $\boldsymbol{\tau}_p^n$ is the stress tensor of the particle p, V_p^0 is the initial volume of the particle, and $\mathbf{g}$ is the acceleration due to gravity.

Grid-to-Particle (G2P) Transfer. After the grid is updated, the changes in velocity and momentum are transferred back to the particles. The particle velocity is updated using the grid velocities and interpolation weights:

$$\mathbf{v}_p^{n+1} = \sum_i \mathbf{v}_i^{n+1} w_{ip}^n,$$

and the new position of the particle is given by:

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

Additionally, the affine velocity field gradient $\mathbf{C}_p^{n+1}$ and deformation gradient $\mathbf{F}_p^{n+1}$ are updated as:

$$\mathbf{C}_p^{n+1} = \frac{4}{(\Delta x)^2} \sum_i w_{ip}^n \mathbf{v}_i^{n+1} (\mathbf{x}_i - \mathbf{x}_p^n)^T,$$

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

The Material Point Method effectively combines Lagrangian (particle-based) and Eulerian (grid-based) approaches, making it highly suitable for simulating materials that experience large deformations, fractures, and complex interactions.

G. Ethical Statement

We confirm that all data used in this study were obtained and utilized in compliance with ethical standards. All participants provided consent, or the data were sourced from publicly available datasets with proper permissions. The use and publication of these data and models pose no societal or ethical harm. Necessary precautions were taken to respect individual rights, including privacy and ethical research principles.

References

[1] Ho Kei Cheng, Seoung Wug Oh, Brian Price, Alexander Schwing, and Joon-Young Lee. Tracking anything with decoupled video segmentation. In *Proc. of IEEE Intl. Conf. on Computer Vision*, pages 1316–1326, 2023. 1

[2] Tianyu Huang, Yihan Zeng, Hui Li, Wangmeng Zuo, and Rynson WH Lau. DreamPhysics: Learning physical properties of dynamic 3d gaussians with video diffusion priors. *arXiv preprint arXiv:2406.01476*, 2024. 1

[3] Bernhard Kerbl, Georgios Kopanas, Thomas Leimkühler, and George Drettakis. 3d gaussian splatting for real-time radiance field rendering. *ACM Trans. Graph.*, 42(4):139–1, 2023. 1

[4] Alexander Kirillov, Eric Mintun, Nikhila Ravi, Hanzi Mao, Chloe Rolland, Laura Gustafson, Tete Xiao, Spencer Whitehead, Alexander C Berg, Wan-Yen Lo, et al. Segment anything. In *Proc. of IEEE Intl. Conf. on Computer Vision*, pages 4015–4026, 2023. 1

[5] Fangfu Liu, Hanyang Wang, Shunyu Yao, Shengjun Zhang, Jie Zhou, and Yueqi Duan. Physics3D: Learning physical properties of 3d gaussians via video diffusion. *arXiv preprint arXiv:2406.04338*, 2024. 1

[6] Shilong Liu, Zhaoyang Zeng, Tianhe Ren, Feng Li, Hao Zhang, Jie Yang, Chunyuan Li, Jianwei Yang, Hang Su, Jun Zhu, et al. Grounding dino: Marrying dino with grounded pre-training for open-set object detection. *arXiv preprint arXiv:2303.05499*, 2023. 1

[7] Tianhe Ren, Shilong Liu, Ailing Zeng, Jing Lin, Kunchang Li, He Cao, Jiayu Chen, Xinyu Huang, Yukang Chen, Feng Yan, et al. Grounded sam: Assembling open-world models for diverse visual tasks. *arXiv preprint arXiv:2401.14159*, 2024. 1

[8] Roman Suvorov, Elizaveta Logacheva, Anton Mashikhin, Anastasia Remizova, Arsenii Ashukha, Aleksei Silvestrov, Naejin Kong, Harshith Goka, Kiwoong Park, and Victor Lempitsky. Resolution-robust large mask inpainting with fourier convolutions. In *Proc. of IEEE Conf. on Computer Vision and Pattern Recognition*, pages 2149–2159, 2022. 1

[9] Jiaxiang Tang, Zhaoxi Chen, Xiaokang Chen, Tengfei Wang, Gang Zeng, and Ziwei Liu. LGM: Large multi-view gaussian model for high-resolution 3d content creation. In *Proc. of European Conf. on Computer Vision*, pages 1–18, 2025. 1

[10] Mingqiao Ye, Martin Danelljan, Fisher Yu, and Lei Ke. Gaussian grouping: Segment and edit anything in 3d scenes. *arXiv preprint arXiv:2312.00732*, 2023. 1

[11] Tianyuan Zhang, Hong-Xing Yu, Rundi Wu, Brandon Y Feng, Changxi Zheng, Noah Snavely, Jiajun Wu, and William T Freeman. PhysDreamer: Physics-based interaction with 3d objects via video generation. *arXiv preprint arXiv:2404.13026*, 2024. 1

[12] Youcai Zhang, Xinyu Huang, Jinyu Ma, Zhaoyang Li, Zhaochuan Luo, Yanchun Xie, Yuzhuo Qin, Tong Luo, Yaqian Li, Shilong Liu, et al. Recognize anything: A strong image tagging model. In *Proc. of IEEE Conf. on Computer Vision and Pattern Recognition*, pages 1724–1732, 2024. 1

[13] Haoyu Zhao, Chen Yang, Hao Wang, Xingyue Zhao, and Wei Shen. SG-GS: Photo-realistic animatable human avatars with semantically-guided gaussian splatting. *arXiv preprint arXiv:2408.09665*, 2024. 1