

PUGS: Zero-shot Physical Understanding with Gaussian Splatting

Yinghao Shuai¹, Ran Yu², Yuantao Chen³, Zijian Jiang¹, Xiaowei Song¹, Nan Wang¹,
Jv Zheng⁴, Jianzhu Ma⁴, Meng Yang⁴, Zhicheng Wang¹, Wenbo Ding², and Hao Zhao^{†4}

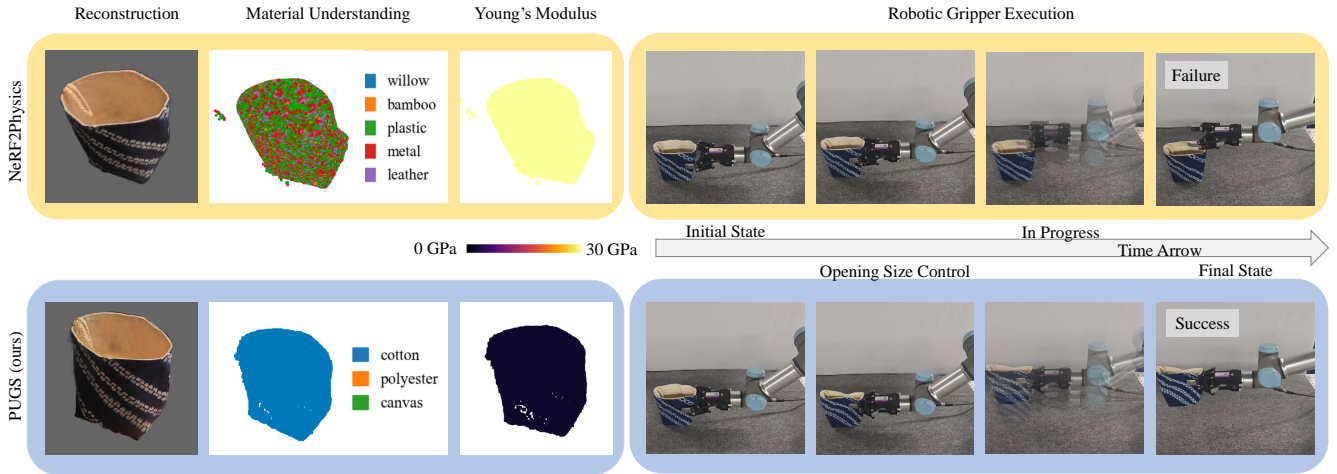


Fig. 1. Comparison between NeRF2Physics [1] and our proposed PUGS. The input is a set of RGB images, and the output is a reconstructed target with physical property understanding. PUGS switches from NeRF to Gaussian Splatting and correctly predicts both the material (cotton) and the Young’s modulus, while NeRF2Physics fails. With the correct Young’s modulus, the robotic gripper can adjust its opening size and successfully grasp the object.

Abstract—Current robotic systems can understand the categories and poses of objects well. But understanding physical properties like mass, friction, and hardness, in the wild, remains challenging. We propose a new method that reconstructs 3D objects using the Gaussian splatting representation and predicts various physical properties in a zero-shot manner. We propose two techniques during the reconstruction phase: a geometry-aware regularization loss function to improve the shape quality and a region-aware feature contrastive loss function to promote region affinity. Two other new techniques are designed during inference: a feature-based property propagation module and a volume integration module tailored for the Gaussian representation. Our framework is named as zero-shot physical understanding with Gaussian splatting, or PUGS. PUGS achieves new state-of-the-art results on the standard benchmark of ABO-500 mass prediction. We provide extensive quantitative ablations and qualitative visualization to demonstrate the mechanism of our designs. We show the proposed methodology can help address challenging real-world grasping tasks. Our codes, data, and models are available at <https://github.com/EverNorif/PUGS>.

I. INTRODUCTION

The current robotic vision systems have made significant strides in recognizing object categories [2] [3] and poses

†Corresponding author.

¹CAD Research Center, Tongji University, China, {shuaiyinghao,jiangzj,kevin729,2233017,zhichengwang}@tongji.edu.cn

²Shenzhen Ubiquitous Data Enabling Key Lab, Shenzhen International Graduate School, Tsinghua University, Shenzhen 518055, China, yur23@mails.tsinghua.edu.cn and ding.wenbo@sz.tsinghua.edu.cn

³The Chinese University of Hong Kong, Shenzhen, China, 224015065@link.cuhk.edu.cn

⁴Institute for AI Industry Research (AIR), Tsinghua University, China, zhaohao@air.tsinghua.edu.cn

[4] [5]. However, for effective grasping and manipulation in unstructured environments, it is crucial to also understand key physical properties of objects. For instance, as shown in Fig. 1, a cotton bag is soft, and if the planning algorithm selects a large gripper opening size, the bag could slip, resulting in a failed grasp. Similarly, understanding an object’s mass and friction is essential for robotic grippers to handle heavy or slippery items. Therefore, this paper focuses on reconstructing objects while inferring their physical properties in a zero-shot fashion.

We build on the framework of NeRF2Physics [1], which takes a set of RGB images as input, reconstructs the target object, and densely predicts its physical properties. Drawing inspiration from the recent success of 3D Gaussian Splatting (3DGS) [6], known for its efficiency and high-quality rendering, we incorporate 3DGS into this task and develop the PUGS framework—standing for zero-shot Physical Understanding with Gaussian Splatting. We identify two key limitations of the vanilla 3DGS: (1) A lack of geometric regularization, where floating Gaussian primitives may blend effectively in RGB volume rendering but introduce difficulties in volume integration for physical properties like mass. (2) A lack of local affinity, as demonstrated in the top-left panel of Fig. 1, where material understanding results are fragmented, making them impractical for downstream tasks such as grasping and manipulation planning.

To address these two issues, we introduce a geometry-aware regularization loss and a region-aware feature contrastive loss. The geometry-aware regularization loss ensures

that the 3D Gaussians align more closely with the object’s spatial structure. The region-aware feature contrastive loss guides the feature vectors of the reconstructed Gaussians to reflect the local region affinity observed in RGB images.

Building on the reconstruction results, we integrate the Vision-Language Foundation Model (VLM) to perform zero-shot physical property prediction. These predictions are propagated to all Gaussians using region-aware features, resulting in dense reconstructions with physical properties. Given the importance of object-level physical properties, such as mass, we propose a Gaussian-based volume integration module. We evaluate PUGS on the mass prediction benchmark introduced by NeRF2Physics [1] and achieve new state-of-the-art (SOTA) results. Visualizations further demonstrate that PUGS provides accurate material understanding and reliable physical property predictions.

In brief, our contributions can be summed up in three-fold:

- We present PUGS, which exploits 3D Gaussian Splatting for efficient object reconstruction and zero-shot physical property inference from RGB images.
- To address limitations in 3DGS, we propose geometry-aware regularization and region-aware feature contrastive losses, improving alignment with object geometry and capturing local region affinities.
- We incorporate Vision-Language Models for zero-shot physical property prediction and introduce a Gaussian-based volume integration module for object-level properties like mass, achieving state-of-the-art results.

II. RELATED WORKS

A. 3D Reconstruction

Recently, Neural Radiance Field (NeRF) [7] has raised a lot of attention due to its photo-realistic rendering ability. 3D Gaussian Splatting (3DGS) [6] goes one step further to achieve real-time rendering. However, most of these works focus on delivering better visual quality but overlook the importance of physical properties, which is critical for application in robotics. Some previous works focus on the accurate geometry [8]–[11], in which PGSR [11] adds a new geometry loss that promises normal-depth consistency. NeRF2Physics [1] utilizes NeRF [7] as a 3D representation and proposes to predict its physical property using Large Language Model (LLM) for the first time. Yet it tries to extract point clouds from NeRF as the scene’s geometry to calculate mass, which is slow and may not be accurate enough.

B. Physical Property Prediction

Physical property prediction from visual data is very important for robotics [12]–[14]. However, collecting paired data between images and various physical properties is very hard. Some previous works propose reasoning physical properties by observing the object’s movement or interaction with other objects in a 3D physical engine [15]–[18]. However, these methods are still limited to a few physical properties and are hard to use. NeRF2Physics [1] is the first to utilize LLMs for physical property prediction tasks in a zero-shot

manner and Octopi [19] further proves its importance in grasping tasks. We aim to make the process faster and more accurate by using 3DGS and a more reliable framework.

C. Vision Language Model

Vision Language Models (VLMs) [20]–[22] have become increasingly popular in robotics, enabling applications across various tasks [23]–[30]. While some methods [23], [30] utilized text and images as input to plan complex tasks, others, like [27], [29], employ CLIP [20] to establish a feature distillation field for aligning 3D representations with text. In our work, We fully leverage the capabilities of VLMs. By utilizing GPT-4 for single-image physical property prediction and CLIP [20] for mapping these properties to 3D reconstructions, we achieve zero-shot physical property prediction with 3D reconstruction.

III. METHOD

A. Overview

As shown in the Fig. 2, our proposed PUGS can reconstruct the object from the calibrated multi-view images \mathcal{I} and subsequently predict the multiple physical properties in a zero shot manner. PUGS can be divided into three phase: shape aware 3DGS reconstruction (Sec. III-B), VLM based physical property prediction (Sec. III-C) and feature based property propagation (Sec. III-D). Firstly, we reconstruct the object using 3DGS, with the geometry-aware regularization loss employed to ensure consistency between the Gaussian and the actual spatial shape distribution of the object. At the same time, we utilize SAM [31] and contrast learning to train region-aware features. Subsequently, the common-sense capabilities and prior knowledge of VLM are employed to predict the physical properties, which are then distilled into the reconstruction results through the combination of CLIP features. Finally, we introduce a way to compute object-level properties based on Gaussian volume integration (Sec. III-E).

B. Shape Aware 3DGS Reconstruction

Preliminary of 3D Gaussian Splatting. 3DGS [6] represents a novel approach for scene representation that not only delivers high-quality rendering in a relatively short training period but also supports real-time rendering and novel view synthesis. Given calibrated multi-view images \mathcal{I} , 3DGS explicitly reconstructs a scene with a realistic rendering result utilizing a set of 3D Gaussians $\{\mathcal{G}_i\}$. Each Gaussian can be represented as follows:

$$\mathcal{G}_i(\mathbf{x}|\mu_i, \Sigma_i) = e^{-\frac{1}{2}(\mathbf{x}-\mu_i)^T \Sigma_i^{-1}(\mathbf{x}-\mu_i)} \quad (1)$$

where $\mu_i \in \mathbb{R}^3$ is the center position of each Gaussian and $\Sigma_i \in \mathbb{R}^{3 \times 3}$ is the corresponding 3D covariance matrix, which can be decomposed into a combination of rotation matrix \mathbf{R} and scaling matrix \mathbf{S} .

$$\Sigma_i = \mathbf{R}_i \mathbf{S}_i \mathbf{S}_i^T \mathbf{R}_i^T \quad (2)$$

Additionally, each 3D Gaussian has an opacity σ and spherical harmonic coefficient SH, which is used to characterise viewpoint-dependent colors.

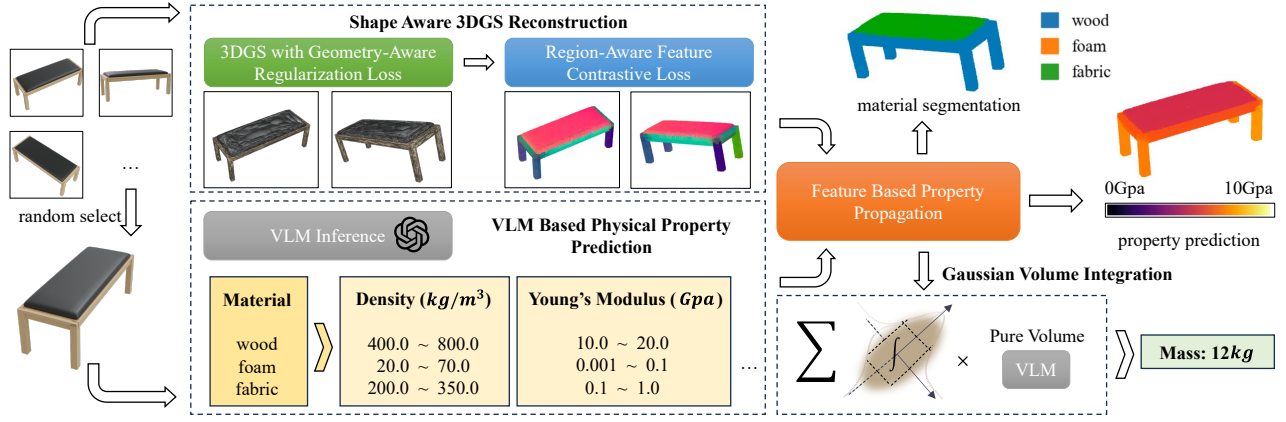


Fig. 2. **Overview of PUGS.** We take the multi-view images of the object as input, reconstruct the Gaussians representation with regional features through Shape Aware 3DGS Reconstruction, predict the candidate properties in a zero-shot manner through VLM Based Physical Property Prediction, and finally obtains the material segmentation result and property prediction result through Feature Based Property Propagation. With proposed Gaussian Volume Integration, we can calculate the object-level property, like mass.

3DGS uses fast α -blending for rendering. Given a transformation W , an intrinsic K and Jacobian matrix J , μ_i and Σ_i can be transformed to camera coordinate and then projected to NDC space. α -blending use the following formula to get the color C of each pixel:

$$C = \sum_{i \in N} T_i \alpha_i c_i, \quad T_i = \prod_{j=1}^{i-1} (1 - \alpha_j) \quad (3)$$

where α_i is calculated with the related learnable opacity σ , c_i is the view-dependent color.

With the rendered images, the loss function is the \mathcal{L}_1 loss and SSIM loss with the GT images:

$$\mathcal{L}_{3dgs} = (1 - \lambda)\mathcal{L}_1 + \lambda\mathcal{L}_{SSIM} \quad (4)$$

Geometry-Aware Regularization Loss. The vanilla 3DGS only uses RGB loss as supervision, which can easily lead to local optima. This results in the distribution of 3D Gaussian being less compatible with the actual shape of the object, which ultimately leads to poor geometry. Our goal is to reconstruct the object with physical understanding and serve it in the downstream tasks of simulation environment, which impose higher demands on the geometry of 3D Gaussian. To address this challenge, we introduce the geometry loss in [11]:

$$\mathcal{L}_{geo} = \sum_{I \in \mathcal{I}} \frac{1}{W(I)} \sum_{p \in W(I)} |\bar{\nabla} I|^5 \|\mathbf{N}_d(p) - \mathbf{N}(p)\|_1 \quad (5)$$

where $W(I)$ is the set of pixel in image I , $\bar{\nabla} I$ is the image gradient normalized to the range of 0 to 1, $\mathbf{N}_d(p)$ is calculated by local plane of the pixel point p and $\mathbf{N}(p)$ is calculated with unbiased depth rendering [11]. Geometry loss can maintain consistency between depth and normal geometry, providing fairly accurate geometric information.

Additionally, we use sparse loss [32] to encourage Gaussian sphere’s opacity value σ to approach either 0 or 1, which helps encourage the distribution of 3D Gaussian to be more



Fig. 3. **Comparison of reconstruction results before and after applying geometry-aware regularization loss (GARL).** Result (a) without the GARL exhibit some floaters and blurred areas; result (b) with the GARL can achieve the results with more accurate geometry.

consistent with the actual spatial shape of the object:

$$\mathcal{L}_{sparse} = \frac{1}{|\sigma|} \sum_{\sigma} [\log(\sigma_i) + \log(1 - \sigma_i)] \quad (6)$$

Finally, the total loss of Gaussian training includes image loss and geometry-aware regularization loss composed of \mathcal{L}_{geo} and \mathcal{L}_{sparse} :

$$\mathcal{L} = \mathcal{L}_{3dgs} + \lambda_1 \mathcal{L}_{geo} + \lambda_2 \mathcal{L}_{sparse} \quad (7)$$

Region-Aware Feature Contrastive Loss. An object be usually composed of a variety of materials, which are typically associated with specific regions. As illustrated in the Fig. 5(a), different regions represent distinct materials. The efficacy of material prediction can be enhanced by accurately identifying the different regions of an object in the reconstruction results, and then lead to an improvement of physical property prediction. In order to achieve this, we follow SAGA [33] by refining the segmentation capability of SAM [31] from 2D Masks to additional features in Gaussian.

Given the calibrated multi-view images, we firstly reconstruct an initial Gaussian representation using RGB loss and geometry-aware regularization loss. Subsequently, we introduce a region-aware feature $f \in \mathbb{R}^D$ for each Gaussian, where D is the dimension of feature. Following the similar

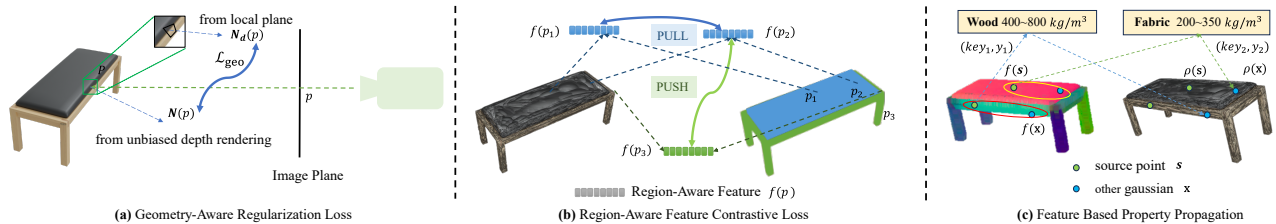


Fig. 4. **Explanation of different modules in PUGS.** During the reconstruction process, we compute the (a) geometry-aware regularization loss using normals obtained through two different methods. With (b) region-aware feature contrastive loss, we pull the features corresponding to Gaussians belonging to the same mask, while pushing apart the features corresponding to different masks. During (c) feature based property propagation, we use the similarity of region-aware feature to propagate physical properties.

approach of α -blending (Eq. 3), the features can be rendered into a 2D feature map \mathbf{F} .

We utilize contrastive learning to train this feature, enabling it to develop region-aware capabilities. We use SAM to extract the mask maps from multi-view images. As illustrated in the Fig. 4(b), if two pixels belong to the same mask, the corresponding features should have a higher similarity. Therefore, the features can be trained with the following contrastive learning loss:

$$\begin{aligned} \mathcal{L}_{\text{coor}}(p_1, p_2) &= [1 - 2\text{coor}(p_1, p_2)] \max[\text{coor}_f(p_1, p_2), 0], \\ \text{coor}_f(p_1, p_2) &= \langle \mathbf{F}(p_1), \mathbf{F}(p_2) \rangle \end{aligned} \quad (8)$$

where $\text{coor}(p_1, p_2)$ indicates whether the pixel p_1 and p_2 belong to the same mask, and $\text{coor}_f(p_1, p_2)$ is defined as the cosine similarity between corresponding features of p_1 and p_2 . Following the SAGA implementation, we also use the feature norm regularization, and the final loss of the region-aware feature in one training image I is:

$$\begin{aligned} \mathcal{L}_{\text{region}} &= \sum_{(p_1, p_2) \in W(I) \times W(I)} \mathcal{L}_{\text{coor}}(p_1, p_2) \\ &+ \frac{1}{HW} \sum_{p \in W(I)} (1 - \|\mathbf{F}(p)\|_2) \end{aligned} \quad (9)$$

In this stage, we only train the region-aware feature on Gaussian, without optimizing other parameters. After this stage of self-supervised training, the region-aware feature on Gaussian is capable of representing regions of objects. If two Gaussian represent the same region, their corresponding feature will demonstrate a high level of similarity.

C. VLM Based Physical Property Prediction

After reconstruction stage in Sec. III-B, the object can be reconstructed using 3D Gaussian, and regionfeature can help to distinguish different regions. Subsequently, we make predictions regarding the physical properties based on the reconstruction results, including its density, hardness, Young’s modulus, and so on.

Similarly to NeRF2Physics [1], we believe that large model exhibits robust common-sense capabilities that can facilitate zero-shot physical understanding. There are some differences in the specifics between our proposed PUGS and NeRF2Physics.

NeRF2Physics employs a two-stage methodology to achieve the prediction of material and physical properties.

Initially, the VQA model (e.g., BLIP2 [22]) is employed to obtain a description of the image. Subsequently, the image description is feed into the large language model, which is then used to predict the material and physical properties. We observe that a certain degree of information is lost during the image-to-text process. To illustrate, we can find the wrong result about a baskets in the rightmost column of Fig. 5. The VQA model predicts the description is three black baskets with a black handle, but the large language model considers it to be the baskets made of rattan, which results in incorrect material understanding.

In contrast to NeRF2Physics, we employ the VLM directly for the physical understanding. One of the multi-view images \mathcal{I} is randomly selected as the input to the VLM, which is then prompted to perform the image description and the prediction of material and physical properties. Finally, we can get a dictionary of K candidate materials $\mathcal{M} = \{(key_k, y_k)\}$, where key_k is the material name and y_k is the specific physical property value or value range.

D. Feature Based Property Propagation

In order to integrate the predicted material and physical properties into the reconstruction result, we employ CLIP [34] as a bridge between disparate modalities. By considering the reconstructed 3D Gaussian as point cloud and performing voxel down-sampling, a set of source points \mathcal{S} distributed on the surface of the object can be obtained. For each source point, we project it onto each input view according to the corresponding camera parameters, and use depth test to determine the occlusion. If the source point is visible on image I , the projection point on the image plane is identified as the centre, and a $p \times p$ patch is extracted to get CLIP feature vector. Ultimately, the CLIP feature \mathbf{z} of each source point will be the average of the CLIP features on different images. With the CLIP feature of the source point, the predicted material and physical properties are fused to the corresponding Gaussian. This is achieved by the following formula:

$$\rho(\mathbf{s}) = \frac{\sum_{k=1}^K \exp(\omega_k[\mathbf{s}]/T) y_k}{\sum_{k=1}^K \exp(\omega_k[\mathbf{s}]/T)} \quad (10)$$

where $\rho(\mathbf{s})$ is the physical property value of source point \mathbf{s} , $\omega_k[\mathbf{s}] = \phi_{\text{CLIP}}(\mathbf{z}, key_k)$ is the cosine similarity between CLIP feature of source point \mathbf{s} and the language CLIP feature of material name key_k , T is a temperature parameter.

After that, we employ interpolation to propagate the physical property from the source point. NeRF2Physics [1] utilizes nearest-neighbor interpolation. However, this approach may result in a more dispersed and fragmented outcome for the material segmentation, as illustrated in Fig. 5(d). In contrast, PUGS employs a interpolation based on the similarity of region-aware feature as illustrated in the Fig. 4(c):

$$\rho(\mathbf{x}) = \rho(\operatorname{argmax}_{\mathbf{s} \in \mathcal{S}} \langle f(\mathbf{x}), f(\mathbf{s}) \rangle) \quad (11)$$

where $f(\mathbf{x})$ and $f(\mathbf{s})$ are the region-aware feature of Gaussian \mathbf{x} and \mathbf{s} respectively. As shown in Fig. 5(c), our interpolation based on region-aware feature enables the attainment of more uniform and precise material segmentation results.

E. Gaussian Volume Integration

Following NeRF2Physics [1], we also consider the prediction of object-level physical properties, such as mass, which requires the estimation and integration of the volume of an object. In contrast to the approach taken in NeRF2Physics, where volume estimation is based on the predicted material thickness, we employ the explicit properties of 3D Gaussian for the integration. Each 3D Gaussian is a 3D ellipsoid in the space, and the corresponding opacity σ can be used to weight it. The associated point-level physical properties are then multiplied by the weight, and perform a cumulative computation to obtain the object-level physical properties.

Benefit from the geometry loss in Sec. III-B, the reconstructed 3D Gaussian is distributed along the surface of the object. However, the reconstructed objects with 3D Gaussian will exhibit substantial internal voids, so the estimated result is the volume of the surface layer of the object, which still differs from the actual volume. In order to compensate for this discrepancy, we introduce the concept of pure volume. The pure volume of an object is defined as the volume it occupies after the removal of all hollow areas. Consequently, the pure volume of an object is typically smaller than the spatial volume that is commonly perceived. Only when an object is completely solid, its spatial volume equals to the pure volume. Furthermore, the concept of pure volume provides the insight into the scale of the object, which enhances the precision of our predictions regarding object-level property.

We prompt the VLM to get the pure volume v of one object. The specific prompts can be referenced from the implementation details. Then we use it to revise our prediction and get the final prediction result m :

$$m = \frac{v}{c} \hat{m} \quad (12)$$

where \hat{m} is the predicted object-level physical property in the above step, c is the predicted volume of all 3D Gaussian.

IV. EXPERIMENTS

Firstly, we evaluate PUGS on the ABO-500 dataset, which serves as the benchmark for object mass estimation in NeRF2Physics [1]. Subsequently, PUGS is adapted for downstream applications, enabling the robotic arm to perform object grasping based on the predicted physical property.

TABLE I
QUANTITATIVE RESULTS OF OBJECT MASS ESTIMATION ON ABO-500.

Approach	ADE (\downarrow)	ALDE (\downarrow)	APE (\downarrow)	MnRE (\uparrow)
NeRF2Physics	12.725	0.736	1.040	0.564
PUGS (Ours)	9.461	0.661	0.767	0.576
	+25.65%	+10.19%	+26.25%	+2.13%

A. Object Mass Estimation

Dataset. We utilize the ABO-500 dataset for object mass estimation. The ABO-500 is a subset derived from the Amazon Berkeley Objects (ABO) dataset [35], initially used by NeRF2Physics. The ABO-500 subset specifically includes 500 objects from ABO dataset, each accompanied by calibrated multi-view images and object mass annotations.

Metrics. Follow NeRF2Physics, we also use the same metrics for evaluation, including Absolute Difference Error (ADE), Absolute Log Difference Error (ALDE), Absolute Percentage Error (APE) and Min Ratio Error (MnRE).

Qualitative Results. Fig. 5 presents a comparison of material segmentation results between our PUGS and NeRF2Physics. It can be observed that our pipeline produces more coherent and reliable material segmentation with fewer fragmented results.

Quantitative Results. We report the mass estimation metrics on the whole ABO-500 in Tab. I, where bold number means better result. The result shows that our pipeline achieved superior results across all evaluated metrics.

Ablations. We perform ablations with different approach settings. Firstly, we remove the geometry-aware regularization loss. The results indicate that geometry-aware regularization loss provides a modest improvement. Fig. 3 shows that geometry-aware regularization loss enhances the geometric structure of objects with fine details. Therefore, we select 100 objects (subset A) from the ABO-500 dataset that possess more intricate structures, such as shelves with mesh designs. As shown in the Tab. II, a greater average improvement can be observed on these objects. Next, we remove the region-aware feature training. Since the majority of objects are dominated by a single material, removing it results in only a slight performance drop. However, as shown in the Fig. 5, it helps produce more reliable material segmentation. We also select 100 objects (subset B) composed of multiple materials. As indicated in the Tab. III, region-aware feature leads to a more significant improvement for these objects. Lastly, we change our Gaussian volume integration to thickness based integration from NeRF2Physics [1]. The results indicate that the thickness-based volume integration method is not suitable for our Gaussian representations. It is difficult to find a single appropriate correction factor for all objects.

B. Application

We apply the reconstructed results with physical properties from PUGS to object grasping tasks. Firstly, we collect videos of the objects from real-world scenarios using a



Fig. 5. **Region Aware Feature Visualization and Material Segmentation Result of some object.** The visualization results in (b) demonstrate that our region-aware feature can effectively identify different regions of the object. (c) and (d) represent the material prediction results of PUGS and NeRF2Physics, respectively. The results from NeRF2Physics are more fragmented, whereas our PUGS achieves more coherent and accurate material segmentation.

TABLE II

ABLATION ABOUT GEOMETRY-AWARE REGULARIZATION LOSS (**GARL**).

Approach Setting	ADE (\downarrow)	ALDE (\downarrow)	APE (\downarrow)	MnRE (\uparrow)
w/o GARL (500 objects)	9.804	0.675	0.858	0.567
w/ GARL (500 objects)	9.461	0.661	0.767	0.576
	+3.50%	+2.07%	+10.60%	+1.59%
w/o GARL (subset A)	5.332	0.751	1.340	0.514
w/ GARL (subset A)	4.218	0.652	1.124	0.568
	+20.89%	+13.18%	+16.12%	+10.51%

TABLE III

ABLATION ABOUT REGION AWARE FEATURE TRAINING (**RAFT**).

Approach Setting	ADE (\downarrow)	ALDE (\downarrow)	APE (\downarrow)	MnRE (\uparrow)
w/o RAFT (500 objects)	9.625	0.669	0.812	0.571
w/ RAFT (500 objects)	9.461	0.661	0.767	0.576
	+1.70%	+1.20%	+5.54%	+0.87%
w/o RAFT (subset B)	6.302	0.684	1.224	0.555
w/ RAFT (subset B)	5.602	0.634	1.117	0.583
	+11.17%	+7.30%	+8.74%	+5.06%

smartphone. These videos are then processed to extract frames and get multi-view images. Then we use COLMAP to estimate the camera parameters. Based on the calibrated multi-view images, PUGS are employed to reconstruct these objects and predict their Young’s modulus, allowing us to assess the deformability of the objects. Since NeRF2Physics [1] can also predict various physical properties, we conducted a comparative analysis with the results produced by it.

As shown in the Fig. 1, we conduct an experiment on a cotton package. The opening size of robotic gripper can be

TABLE IV

ABLATION ABOUT **THICKNESS** AND **GAUSSIAN** BASED INTEGRATION

Approach Setting	ADE (\downarrow)	ALDE (\downarrow)	APE (\downarrow)	MnRE (\uparrow)
THICKNESS (500 objects)	12.498	0.811	1.803	0.537
GAUSSIAN (500 objects)	9.461	0.661	0.767	0.576
	+24.30%	+18.50%	+57.46%	+7.26%

adjusted based on the predicted Young’s modulus. A higher Young’s modulus indicates lower deformability, requiring a larger opening size. NeRF2Physics [1] incorrectly predicted the Young’s modulus of the object (30+ GPa), resulting in an opening size nearly equal to the object’s width, which ultimately leads to a failed grasp. In contrast, PUGS accurately predicted the physical property (0.5+ GPa), enabling the gripper to successfully grasp the object.

V. CONCLUSION

In this paper, we propose PUGS, a 3DGS-based framework capable of densely reconstructing for object’s physical properties from multi-view images. Experimental results and visualizations demonstrate that PUGS achieves accurate and coherent predictions of material and physical properties, while also benefiting downstream tasks in robotics. Future work includes expanding the framework to enable physical understanding at the scene level.

REFERENCES

- [1] A. J. Zhai, Y. Shen, E. Y. Chen, G. X. Wang, X. Wang, S. Wang, K. Guan, and S. Wang, "Physical property understanding from language-embedded feature fields," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2024, pp. 28 296–28 305.
- [2] T. Cheng, L. Song, Y. Ge, W. Liu, X. Wang, and Y. Shan, "Yolo-world: Real-time open-vocabulary object detection," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2024, pp. 16 901–16 911.
- [3] X. Chen, H. Zhao, G. Zhou, and Y.-Q. Zhang, "Pq-transformer: Jointly parsing 3d objects and layouts from point clouds," *IEEE Robotics and Automation Letters*, vol. 7, no. 2, pp. 2519–2526, 2022.
- [4] B. Wen, W. Yang, J. Kautz, and S. Birchfield, "Foundationpose: Unified 6d pose estimation and tracking of novel objects," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2024, pp. 17 868–17 879.
- [5] L. Zhong, Y. Zhang, H. Zhao, A. Chang, W. Xiang, S. Zhang, and L. Zhang, "Seeing through the occluders: Robust monocular 6-dof object pose tracking via model-guided video object segmentation," *IEEE Robotics and Automation Letters*, vol. 5, no. 4, pp. 5159–5166, 2020.
- [6] B. Kerbl, G. Kopanas, T. Leimkühler, and G. Drettakis, "3d gaussian splatting for real-time radiance field rendering," *ACM Transactions on Graphics*, vol. 42, no. 4, July 2023. [Online]. Available: <https://repo-sam.inria.fr/fungraph/3d-gaussian-splatting/>
- [7] B. Mildenhall, P. P. Srinivasan, M. Tancik, J. T. Barron, R. Ramamoorthi, and R. Ng, "Nerf: Representing scenes as neural radiance fields for view synthesis," in *ECCV*, 2020.
- [8] A. Guédon and V. Lepetit, "Sugar: Surface-aligned gaussian splatting for efficient 3d mesh reconstruction and high-quality mesh rendering," *CVPR*, 2024.
- [9] B. Huang, Z. Yu, A. Chen, A. Geiger, and S. Gao, "2d gaussian splatting for geometrically accurate radiance fields," in *SIGGRAPH 2024 Conference Papers*. Association for Computing Machinery, 2024.
- [10] B. Zhang, C. Fang, R. Shrestha, Y. Liang, X. Long, and P. Tan, "Rade-gs: Rasterizing depth in gaussian splatting," 2024. [Online]. Available: <https://arxiv.org/abs/2406.01467>
- [11] D. Chen, H. Li, W. Ye, Y. Wang, W. Xie, S. Zhai, N. Wang, H. Liu, H. Bao, and G. Zhang, "Pgsr: Planar-based gaussian splatting for efficient and high-fidelity surface reconstruction," 2024.
- [12] K. Zhang, B. Li, K. Hauser, and Y. Li, "Adaptigraph: Material-adaptive graph-based neural dynamics for robotic manipulation," in *Proceedings of Robotics: Science and Systems (RSS)*, 2024.
- [13] H. Shi, H. Xu, S. Clarke, Y. Li, and J. Wu, "Robocook: Long-horizon elasto-plastic object manipulation with diverse tools," in *Conference on Robotic Learning (CoRL)*, 2023.
- [14] Y. Li, J. Wu, R. Tedrake, J. B. Tenenbaum, and A. Torralba, "Learning particle dynamics for manipulating rigid bodies, deformable objects, and fluids," in *ICLR*, 2019.
- [15] X. Li, Y.-L. Qiao, P. Y. Chen, K. M. Jatavallabhula, M. Lin, C. Jiang, and C. Gan, "PAC-nerf: Physics augmented continuum neural radiance fields for geometry-agnostic system identification," in *The Eleventh International Conference on Learning Representations*, 2023. [Online]. Available: <https://openreview.net/forum?id=tVkrbkz42vc>
- [16] L. Pinto, D. Gandhi, Y. Han, Y.-L. Park, and A. Gupta, "The curious robot: Learning visual representations via physical interactions," in *Computer Vision – ECCV 2016*, B. Leibe, J. Matas, N. Sebe, and M. Welling, Eds. Cham: Springer International Publishing, 2016, pp. 3–18.
- [17] J. Wu, I. Yildirim, J. J. Lim, B. Freeman, and J. Tenenbaum, "Galileo: Perceiving physical object properties by integrating a physics engine with deep learning," in *Advances in Neural Information Processing Systems*, C. Cortes, N. Lawrence, D. Lee, M. Sugiyama, and R. Garnett, Eds., vol. 28. Curran Associates, Inc., 2015. [Online]. Available: https://proceedings.neurips.cc/paper_files/paper/2015/file/d09bf41544a3365a46c9077ebb5e35c3-Paper.pdf
- [18] S. Yao and K. Hauser, "Estimating tactile models of heterogeneous deformable objects in real time," in *2023 IEEE International Conference on Robotics and Automation (ICRA)*, 2023, pp. 12 583–12 589.
- [19] S. Yu, K. Lin, A. Xiao, J. Duan, and H. Soh, "Octopi: Object property reasoning with large tactile-language models," *arXiv preprint arXiv:2405.02794*, 2024.
- [20] G. Ilharco, M. Wortsman, R. Wightman, C. Gordon, N. Carlini, R. Taori, A. Dave, V. Shankar, H. Namkoong, J. Miller, H. Hajishirzi, A. Farhadi, and L. Schmidt, "Openclip," Jul. 2021, if you use this software, please cite it as below. [Online]. Available: <https://doi.org/10.5281/zenodo.5143773>
- [21] D. Zhu, J. Chen, X. Shen, X. Li, and M. Elhoseiny, "Minigpt-4: Enhancing vision-language understanding with advanced large language models," *arXiv preprint arXiv:2304.10592*, 2023.
- [22] J. Li, D. Li, S. Savarese, and S. Hoi, "Blip-2: Bootstrapping language-image pre-training with frozen image encoders and large language models," in *International conference on machine learning*. PMLR, 2023, pp. 19 730–19 742.
- [23] W. Huang, C. Wang, R. Zhang, Y. Li, J. Wu, and L. Fei-Fei, "Voxposer: Composable 3d value maps for robotic manipulation with language models," *arXiv preprint arXiv:2307.05973*, 2023.
- [24] A. Stone, T. Xiao, Y. Lu, K. Gopalakrishnan, K.-H. Lee, Q. Vuong, P. Wohlhart, S. Kirmani, B. Zitkovich, F. Xia, C. Finn, and K. Hausman, "Open-world object manipulation using pre-trained vision-language models," in *Conference on Robotic Learning (CoRL)*, 2023.
- [25] S. Chen, R. Garcia, C. Schmid, and I. Laptev, "Polarnet: 3d point clouds for language-guided robotic manipulation," in *Conference on Robotic Learning (CoRL)*, 2023.
- [26] A. Brohan, N. Brown, J. Carbajal, Y. Chebotar, X. Chen, K. Chormanski, T. Ding, D. Driess, A. Dubey, C. Finn, P. Florence, C. Fu, M. G. Arenas, K. Gopalakrishnan, K. Han, K. Hausman, A. Herzog, J. Hsu, B. Ichter, A. Irpan, N. Joshi, R. Julian, D. Kalashnikov, Y. Kuang, I. Leal, L. Lee, T.-W. E. Lee, S. Levine, Y. Lu, H. Michalewski, I. Mordatch, K. Pertsch, K. Rao, K. Reymann, M. Ryoo, G. Salazar, P. Sanketi, P. Sermanet, J. Singh, A. Singh, R. Soricut, H. Tran, V. Vanhoucke, Q. Vuong, A. Wahid, S. Welker, P. Wohlhart, J. Wu, F. Xia, T. Xiao, P. Xu, S. Xu, T. Yu, and B. Zitkovich, "Rt-2: Vision-language-action models transfer web knowledge to robotic control," in *arXiv preprint arXiv:2307.15818*, 2023.
- [27] M. Qin, W. Li, J. Zhou, H. Wang, and H. Pfister, "Langsplat: 3d language gaussian splatting," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2024, pp. 20 051–20 060.
- [28] S. Zhou, H. Chang, S. Jiang, Z. Fan, Z. Zhu, D. Xu, P. Chari, S. You, Z. Wang, and A. Kadambi, "Feature 3dgs: Supercharging 3d gaussian splatting to enable distilled feature fields," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2024, pp. 21 676–21 685.
- [29] J. Kerr, C. M. Kim, K. Goldberg, A. Kanazawa, and M. Tancik, "Lerf: Language embedded radiance fields," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2023, pp. 19 729–19 739.
- [30] G. Zhou, Y. Hong, and Q. Wu, "Navgpt: Explicit reasoning in vision-and-language navigation with large language models," in *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 38, no. 7, 2024, pp. 7641–7649.
- [31] A. Kirillov, E. Mintun, N. Ravi, H. Mao, C. Rolland, L. Gustafson, T. Xiao, S. Whitehead, A. C. Berg, W.-Y. Lo, P. Dollár, and R. Girshick, "Segment anything," *arXiv:2304.02643*, 2023.
- [32] Q. Xu, Z. Xu, J. Philip, S. Bi, Z. Shu, K. Sunkavalli, and U. Neumann, "Point-nerf: Point-based neural radiance fields," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2022, pp. 5438–5448.
- [33] J. Cen, J. Fang, C. Yang, L. Xie, X. Zhang, W. Shen, and Q. Tian, "Segment any 3d gaussians," *arXiv preprint arXiv:2312.00860*, 2023.
- [34] A. Radford, J. W. Kim, C. Hallacy, A. Ramesh, G. Goh, S. Agarwal, G. Sastry, A. Askell, P. Mishkin, J. Clark, G. Krueger, and I. Sutskever, "Learning transferable visual models from natural language supervision," 2021. [Online]. Available: <https://arxiv.org/abs/2103.00020>
- [35] J. Collins, S. Goel, K. Deng, A. Luthra, L. Xu, E. Gundogdu, X. Zhang, T. F. Yago Vicente, T. Dideriksen, H. Arora, M. Guillaumin, and J. Malik, "Abo: Dataset and benchmarks for real-world 3d object understanding," *CVPR*, 2022.