

# ObjectGS: Object-aware Scene Reconstruction and Scene Understanding via Gaussian Splatting

Ruijie Zhu<sup>1,2\*</sup> Mulin Yu<sup>2</sup> Lining Xu<sup>3</sup> Lihan Jiang<sup>1,2</sup> Yixuan Li<sup>3</sup>  
 Tianzhu Zhang<sup>1†</sup> Jiangmiao Pang<sup>2</sup> Bo Dai<sup>4</sup>

<sup>1</sup> University of Science and Technology of China <sup>2</sup> Shanghai Artificial Intelligence Laboratory  
<sup>3</sup> The Chinese University of Hong Kong <sup>4</sup> The University of Hong Kong

## Abstract

*3D Gaussian Splatting is renowned for its high-fidelity reconstructions and real-time novel view synthesis, yet its lack of semantic understanding limits object-level perception. In this work, we propose ObjectGS, an object-aware framework that unifies 3D scene reconstruction with semantic understanding. Instead of treating the scene as a unified whole, ObjectGS models individual objects as local anchors that generate neural Gaussians and share object IDs, enabling precise object-level reconstruction. During training, we dynamically grow or prune these anchors and optimize their features, while a one-hot ID encoding with a classification loss enforces clear semantic constraints. We show through extensive experiments that ObjectGS not only outperforms state-of-the-art methods on open-vocabulary and panoptic segmentation tasks, but also integrates seamlessly with applications like mesh extraction and scene editing. Project page: [https://ruijiezh94.github.io/ObjectGS\\_page](https://ruijiezh94.github.io/ObjectGS_page)*

## 1. Introduction

3D scene reconstruction and understanding in open-world settings remain challenging yet crucial for applications like embodied AI where robots must recognize and grasp target objects, and film editing, which requires precise 3D object extraction. Recent advances in NeRF [1, 29, 42] and 3D Gaussian Splatting [16, 27, 48, 52] have enabled high-quality reconstructions and real-time rendering, but they lack semantic understanding, hindering direct object extraction. Although 2D Vision Foundation Models [18, 37] excel at instance segmentation, they fail to maintain 3D consistency across views.

To address this issue, recent approaches [4, 8, 33, 45, 47]

\*The work is done during Ruijie Zhu's internship at Shanghai AI Lab.

†Corresponding author.

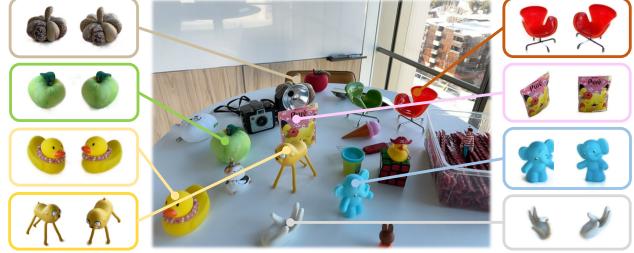


Figure 1. In the open-world setting, ObjectGS enables 3D object awareness during reconstruction, allowing it to achieve high-quality scene reconstruction and understanding simultaneously.

integrate these 2D VFs into 3DGS frameworks, enabling open-vocabulary segmentation in 3D. Although these methods have achieved 3D instance segmentation in open scenes, we identify two overlooked issues. First, some methods [4, 33, 45] treat 3D reconstruction and segmentation as separate tasks, ignoring their inherent interdependence—where precise reconstruction is key to accurate segmentation, and semantic cues help resolve ambiguities. For instance, as shown in Fig. 2(a), it is hard to perform segmentation on a misconstrued Gaussian, but incorporating semantic information during reconstruction can help eliminate such ambiguity. Second, current approaches [23, 33, 47] use continuous 3D semantic fields for segmentation, which contradicts the inherently discrete nature of semantic classification and introduces ambiguity during alpha blending. As shown in Fig. 2(b), regression-based Gaussian semantic features inevitably introduce vagueness in alpha blending.

Building on above analysis, we propose ObjectGS, a Gaussian splatting framework that unifies scene reconstruction and understanding by modeling each object as a collection of Gaussians, as shown in Fig. 1. Specifically, our method consists of three key components: (1) *Object ID Labeling and Voting*: Leveraging a SAM-based segmentation pipeline, we generate consistent semantic labels across

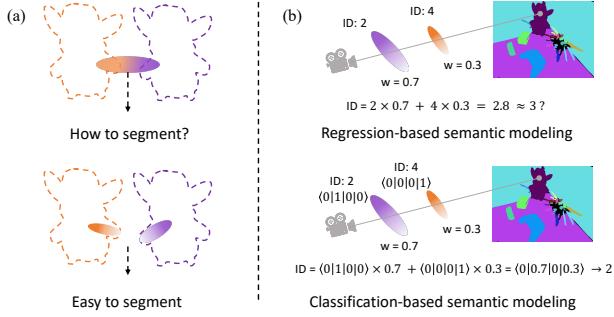


Figure 2. (a) Considering semantic information during reconstruction can help better model objects. (b) Existing semantic modeling methods often lead to semantic ambiguity during alpha blending, whereas our classification-based semantic modeling eliminates this problem by independently accumulating the semantics of different objects through ID encoding.

views and employ a majority voting scheme to robustly assign object IDs to the initial scene point cloud, laying a strong foundation for object differentiation. (2) *Object-aware Neural Gaussian Generation*: Building on these object IDs, we introduce a novel strategy inspired by Scaffold-GS [27] to generate anchors—minimal modeling units enriched with 3DGS [16] or 2DGS [12] primitives—that dynamically grow or prune during reconstruction, ensuring each object’s unique features are accurately captured. (3) *Discrete Gaussian Semantics modeling*: To guarantee unambiguous object recognition, we assign each neural Gaussian a fixed one-hot ID encoding based solely on its object ID, a departure from conventional learnable semantics. This discrete representation enables precise 2D splatting and pixel-level object identification, effectively bridging the gap between reconstruction and semantic understanding.

Our main contributions can be summarized as follows:

- We propose ObjectGS, a novel Gaussian Splatting framework that unifies scene reconstruction and understanding in open-world settings.
- We develop an object-aware training framework that leverages semantic cues to adaptively model objects.
- We introduce a classification-based approach to Gaussian semantics, achieving precise 3D instance segmentation.
- Extensive experiments show that our method outperforms state-of-the-art approaches, while seamlessly supporting scene decomposition and editing.

## 2. Related Work

**3D Gaussian Splatting.** After the tremendous success of Neural Radiance Field (NeRF) [29] in novel view synthesis and 3D reconstruction, 3D Gaussian Splatting (3DGS) [16] has emerged as the new favorite, gaining significant attention from the research community. Compared to NeRF, 3DGS offers explicit scene representation, high-quality re-

construction, and real-time rendering, which presents a broader range of application prospects [14, 15, 26, 36, 51, 56]. Our work is built upon Scaffold-GS [27], with the core idea being the generation of neural Gaussians through anchors, thereby creating a hierarchical scene representation. Furthermore, we extend this framework by modeling the semantics of Gaussians, enabling it to perceive objects in the scene while performing reconstruction. This enhances our ability to simultaneously achieve both 3D scene reconstruction and understanding.

**Open-world 2D Segmentation.** The development of visual foundation models [2, 18, 31, 34, 41] has accelerated the application of low-level visual tasks [6, 13, 20, 25, 38, 53–55, 57]. Among them, 2D segmentation tasks have gradually started to address general scene segmentation. SAM [18] is a milestone in this area, showcasing impressive zero-shot segmentation capabilities in open-world scenarios. It can fulfill specific segmentation needs through flexible prompts, such as points, bounding boxes, or text, and can even perform automatic segmentation without any prompts. However, SAM does not directly enable cross-frame consistency for video segmentation. Subsequent methods [7, 35] extended SAM’s capabilities to unlock the potential for open-world video segmentation. Despite these advances, using only 2D vision foundation models does not directly solve the problem of 3D scene segmentation. As a result, some early methods [3, 9, 19, 30, 40, 46, 50] have begun to explore combining 3D representation models with SAM to lift its capabilities to 3D scene segmentation.

**Open-world 3D Scene Understanding.** With the rise of 3DGS, recent works [4, 8, 23, 28, 32, 33, 47] have started to combine 3DGS with 2D vision foundation models for open-vocabulary scene understanding. For example, Langsplat [33] combines SAM and CLIP to extract object features and constructs a 3D language field on top of 3DGS using the CLIP features of the objects, enabling open-vocabulary 3D object segmentation. Unlike Langsplat, Gaussian Grouping [47] directly leverages DEVA [7] to extract ID-consistent masks across multiple views, which are then used to supervise the identity features of each Gaussian, enabling efficient 3D segmentation and scene editing. By summarizing existing methods, we find that they typically rely on constructing learnable Gaussian semantic features to achieve 3D segmentation. However, due to the inherent sparsity and uniqueness of semantic features, these methods often require additional regularization terms [4, 32, 47] or contrastive losses [4, 8] to mitigate the ambiguity of Gaussian semantics. In contrast, we innovatively propose a new paradigm that constrains deterministic Gaussian semantics to guide object-aware Gaussians to reconstruct their corresponding objects.

### 3. Methodology

The overall architecture of our method is shown in Fig. 3. In Sec. 3.1, we first introduce our data preprocessing pipeline, where we extract ID-consistent object masks and use them to initialize the point clouds for different objects. In Sec. 3.2, we describe how the initialized point cloud generates anchors and their corresponding Gaussians. In Sec. 3.3, to enable Gaussian semantic awareness, we model the semantics of the Gaussians and construct classification-based semantic constraints. In Sec. 3.4, we introduce the training objectives in our method.

#### 3.1. Initialization

To consistently lift the semantic information from powerful visual foundation models into 3D, we first extract object masks with consistent IDs across multiple views and then apply a majority voting strategy to assign these masks to each object in 3D space.

**Object ID Labeling.** Following Gaussian grouping [47], we use DEVA [7] to obtain object masks with ID consistency across multiple views. Additionally, to enable open-vocabulary object queries, we also support text and click prompts for selecting specific target objects, with the help of Grounded-SAM [37]. Given a sequence of images  $\{I_i\}$ , we use this pipeline to obtain the ID corresponding to each object in the scene:

$$\{L_i\} = \text{SAM}(\{I_i\}, \text{Prompts}), \quad (1)$$

where the value of  $L_i$  indicates the object IDs of pixels in image  $I_i$  and the prompts are optional clicks or texts. Each pixel has an object ID. For some unclassified pixels (no predicted category or invalid value), we uniformly define their ID as 0. Assume that there are  $n$  objects in the scene, we assign the object IDs the values  $0, 1, 2, \dots, n$ .

**Object ID Voting.** The initialization of Gaussian splatting framework relies on point clouds. Therefore, we need to assign the IDs of the object masks to the point cloud. We have already noted that there are some methods, such as [11], can segment 3D point clouds aligned with SAM masks. However, for the sake of simplicity and ease of use, we design three kinds of voting strategies to quickly initialize the point cloud for different objects. (1) *Majority Voting*. Given a sequence of images  $\{I_i\}$  with length  $N$ , the corresponding object ID maps  $\{L_i\}$  and a COLMAP point cloud  $P_{3D}$ , we first project the point cloud  $P_{3D}$  to 2D views using the camera poses  $\{C_i\}$ , matching the object ID maps  $\{L_i\}$ . As a result, each 3D point  $P_i$  in  $P_{3D}$  has  $N$  object ID votes from different views. We use the simple majority voting principle to obtain the object ID of 3D point  $P_i$ , thus deriving the updated point cloud  $P_{3D}$  with object IDs. (2) *Probability-based Voting*. Similar to the majority voting, probability-based voting also project point clouds to achieve

object-aware voting. The only difference is that it converts vote counts into probabilities rather than directly taking the majority decision to avoid winner-takes-all situations. (3) *Correspondence-based Voting*. Since the point clouds reconstructed by COLMAP maintain the correspondence between 2D and 3D points, a natural idea is to directly utilize these correspondence as the votes. Therefore, we also try to replace the projecting procedure of the majority voting with the COLMAP correspondence. The detailed procedure of these three voting strategies are shown in Sec. 8 of our supplementary.

#### 3.2. Object-aware Neural Gaussian Generation

After obtaining the point cloud from the voting process, we use the point clouds corresponding to different objects to initialize anchors, which serve as the carriers for generating and controlling Gaussian primitives. Similar to Scaffold-GS [27], each anchor corresponds to the center of a voxelized grid from the point cloud and carries a local context feature, a scaling factor, and  $k$  learnable offsets. Since the initialized anchors may be erroneous or sparse, during training, the anchors adaptively grow and prune in the voxel grid to meet the requirements of scene reconstruction.

**Object-aware Anchors.** To enable object awareness, we add an object ID to each anchor, which refers to the corresponding object in the scene. During the growing process, anchors replicate their object IDs, while pruning removes the object IDs. This design has two main benefits: (1) Anchors for the same object can only be generated by anchors with the same object ID, ensuring that newly generated anchors inherit the features of the same object. (2) Each voxel grid corresponds to at most one anchor and its object ID, ensuring semantic exclusivity and determinism in 3D space. Through this simple yet effective design, we can generate object-aware anchors as the basic semantic units.

**Object-aware Neural Gaussians.** For each anchor, we generate  $k$  neural Gaussian primitives (3DGS/2DGS)<sup>1</sup>. The generated Gaussian primitives can be parameterized by their position  $\mu$ , opacity  $\alpha$ , color  $c$ , scale  $s$ , and quaternion  $q$ . Similar to Scaffold-GS, the Gaussian position can be calculated as:

$$\{\mu_0, \dots, \mu_{k-1}\} = x + \{o_0, \dots, o_{k-1}\} \cdot l, \quad (2)$$

where  $\{o_0, \dots, o_{k-1}\}$  are learnable offsets and  $x, l$  are the center and scaling factor of the anchor. Other Gaussian attributes such as color  $c$  can be computed via an MLP as:

$$\{c_0, \dots, c_{k-1}\} = \text{MLP}(f, \delta, d), \quad (3)$$

where  $f$  is the anchor feature,  $\delta$  and  $d$  are the viewing distance and direction between the camera and anchor point.

<sup>1</sup>In the current implementation, 3DGS and 2DGS primitives cannot co-exist in a single model, so only one of them can be chosen at once. Unless otherwise specified, we use the 3DGS primitive by default in this paper.

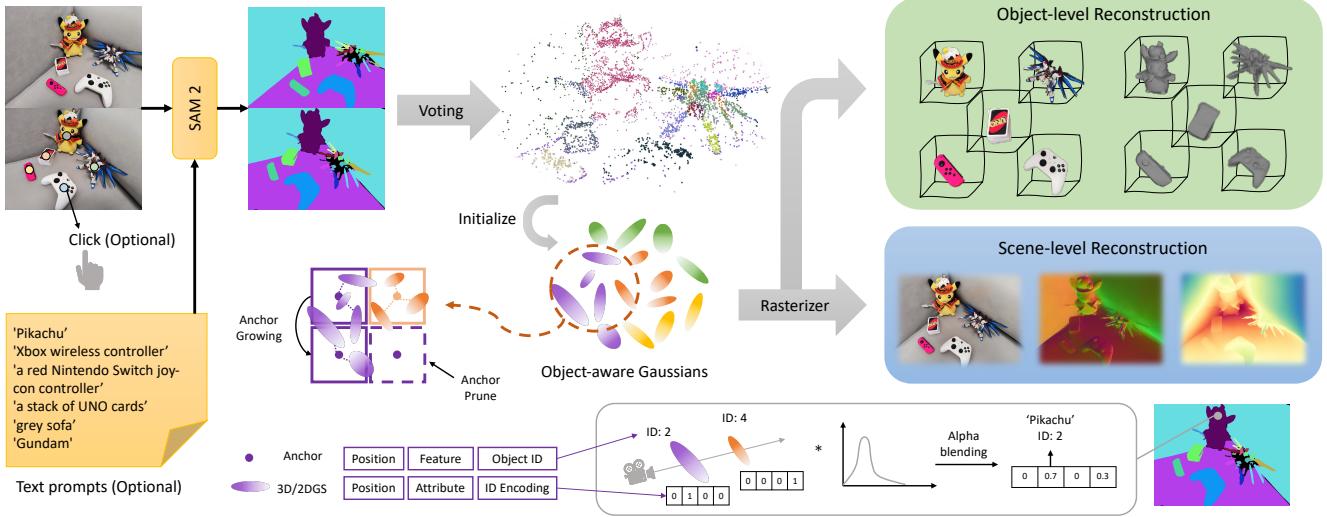


Figure 3. The overall architecture of ObjectGS. We first use a 2D segmentation pipeline to assign object ID and lift it to 3D. Then we initialize the anchors and use them to generate object-aware neural Gaussians. To provide semantic guidance, we model the Gaussian semantics and construct classification-based constraints. As a result, our method enables both object-level and scene-level reconstruction.



Figure 4. Since 2D segmentation method [37] don't account for occluded object, it cannot be used to supervise the independent rendering of objects. In contrast, our ObjectGS render semantics in the scene level, which is occlusion-aware.

### 3.3. Discrete Gaussian Semantic Modeling

In our design, the semantics of the anchors are already modeled using object IDs. A natural idea is to let the generated Gaussian primitives inherit the anchor's object ID, allowing us to easily achieve semantic modeling in 3D space. However, since the object masks are in 2D space, we need to establish a correspondence between 2D and 3D semantics in order to effectively constrain the Gaussian semantics. After analysis, we identify several approaches for this process:

**(a) Learnable Gaussian Semantics.** One simple approach is to follow the color rasterize pipeline to define learnable Gaussian semantic features and optimize them through 2D feature distillation. This approach is widely used in existing methods [4, 5, 23, 32, 33] of Gaussian semantic modeling. While this approach might seem reason-

able, it overlooks a key distinction between the color and semantic attributes of Gaussians: color can be continuous, while semantics are discrete. As mentioned in Fig. 2, blending the semantics via alpha blending in this manner could confuse the Gaussian semantics of different categories and introduce ambiguity.

**(b) Object-independent Constraints.** To resolve the semantic ambiguity in alpha blending, one possible solution is to query the object IDs and independently render objects. By iterating over all object IDs, we can render the object masks for all objects in the scene and use pseudo ground truths derived in 2D segmentation pipeline for supervision. While this approach may seem feasible, it misses a crucial point as shown in Fig. 4: when segmenting objects in 2D images to generate pseudo object mask labels, it don't account for occluded objects. Similar observation is also included in [44]. Therefore, this method cannot handle the complexities of occlusion in scenes where objects overlap.

**(c) One-hot ID Encoding.** To address the above issues, we propose using one-hot ID encoding as the modeling of Gaussian semantics, where the length of the ID encoding is equal to the number of objects in the scene. If there are  $n$  objects, we assign object IDs as  $1, 2, \dots, n$ , and for an object with ID  $i$ , its one-hot encoding vector  $\mathbf{E}_i$  is defined as:

$$\mathbf{E}_i = [0, 0, \dots, 1, \dots, 0]. \quad (\text{with } 1 \text{ in the } i\text{-th dim}) \quad (4)$$

Each anchor has an object ID, and all Gaussians generated by the same anchor share the same one-hot ID encoding.

**Gaussian Semantic Rendering.** During rendering, alpha blending is performed across the Gaussians along the

ray, and the accumulated ID encoding at each pixel is computed as:

$$\mathbf{P}(\mathbf{x}) = \sum_k \alpha_k \cdot T_k \cdot \mathbf{E}_{i_k}, \quad (5)$$

where  $\alpha_k$  and  $T_k$  are the opacity and the accumulated transmittance of the  $k$ -th Gaussian along the ray at pixel  $\mathbf{x}$ ,  $\mathbf{E}_{i_k}$  is the one-hot ID encoding of the  $k$ -th Gaussian with object ID  $i_k$ .  $\mathbf{P}(\mathbf{x})$  is the resulting classification probability vector at pixel  $\mathbf{x}$ , which represents the probability of the pixel belonging to each object ID. Therefore, the predicted object ID of pixels can be derived by taking the index of the maximum classification probability in  $\mathbf{P}(\mathbf{x})$ :

$$\text{ID}(\mathbf{x}) = \arg \max_i (P_i(\mathbf{x})), \quad (6)$$

where  $\text{ID}(\mathbf{x})$  is the predicted object ID for pixel  $\mathbf{x}$ ,  $P_i(\mathbf{x})$  is the classification probability of object ID  $i$  at pixel  $\mathbf{x}$  in the vector  $\mathbf{P}(\mathbf{x})$ .

**Gaussian Semantic Loss.** After deriving the classification probability of the pixel, we can construct a cross entropy loss instead of a L1 loss to constrain the semantics of the Gaussian:

$$\mathcal{L}_{\text{semantic}} = - \sum_{\mathbf{x}} \sum_{i=1}^n \mathbb{1} (\text{ID}'(\mathbf{x}) = i) \cdot \log (P_i(\mathbf{x})), \quad (7)$$

where  $\mathbb{1}$  is the indicator function, which is 1 if the condition is true and 0 otherwise,  $\text{ID}'(\mathbf{x})$  is the ground truth object ID for pixel  $\mathbf{x}$  derived in Sec. 3.1. This approach ensures that the Gaussian semantics for different objects do not interfere with each other during alpha blending. Plus, since we only need to perform alpha blending once at the scene level, this method is occlusion-aware and highly efficient.

**Variable-length Feature Rasterizer.** Although similar semantic modeling methods have been used in NeRF-based approaches [10, 43], current Gaussian-based methods have not yet adopted this kind of semantic modeling. One possible reason for this is that in the original Gaussian CUDA implementation, Gaussian attributes are of fixed length during rasterization. In contrast, to make the ID encoding length adaptable to scenes with different numbers of objects, we implement a variable-length feature alpha blending. As a result, our Gaussian semantic rendering is both convenient and efficient. Consequently, we only need parallelly splatting all the Gaussians in the scene at once to obtain the semantics of all corresponding objects.

### 3.4. Training Objective

With the help of our object-aware Neural Gaussians and discrete Gaussian Semantic Modeling, our method is capable of simultaneously performing object-aware scene reconstruction and 3D scene understanding. Our overall training loss can be expressed as:

$$\mathcal{L} = \mathcal{L}_1 + \lambda_{\text{SSIM}} \mathcal{L}_{\text{SSIM}} + \lambda_{\text{vol}} \mathcal{L}_{\text{vol}} + \lambda_{\text{semantic}} \mathcal{L}_{\text{semantic}}, \quad (8)$$

Table 1. Open-vocabulary segmentation results on LERF-Mask dataset. We follow Gaussian Grouping [47] to test our method.

Model	figurines		ramen		teatime	
	mIoU	mBIoU	mIoU	mBIoU	mIoU	mBIoU
DEVA [7]	46.2	45.1	56.8	51.1	54.3	52.2
LERF [17]	33.5	30.6	28.3	14.7	49.7	42.6
SA3D [3]	24.9	23.8	7.4	7.0	42.5	39.2
LangSplat [33]	52.8	50.5	50.4	44.7	69.5	65.6
GS Grouping [47]	69.7	67.9	77.0	68.7	71.7	66.1
Gaga [28]	<b>90.7</b>	<b>89.0</b>	64.1	61.6	69.3	66.0
ObjectGS(Ours)	88.2	85.2	<b>88.0</b>	<b>79.9</b>	<b>88.9</b>	<b>88.6</b>

where  $\mathcal{L}_1$  and  $\mathcal{L}_{\text{SSIM}}$  are the appearance loss between rendered images and ground truth images,  $\mathcal{L}_{\text{vol}}$  is the volume regularization term in Scaffold-GS [27] and  $\mathcal{L}_{\text{semantic}}$  is the proposed Gaussian semantic loss.

## 4. Experiment

### 4.1. Experimental Setup

**Setting and Datasets.** To comprehensively evaluate the performance of our method in open-world 3D scene understanding tasks, we set up two experimental setups: *open-vocabulary segmentation (OVS)* and *panoptic segmentation*. For OVS, the goal is to segment target objects in an open scene based on given text prompts. We follow Gaussian Grouping [47] to test our method on the LERF-Mask [17] and 3DOVS [24] datasets. For panoptic segmentation, we conduct experiments on the Replica [39] and ScanNet++ [49] datasets. The goal is to perform instance-level segmentation of each object in the scene.

**Implementation Details.** Following the configuration of Scaffold-GS [27], we set the number of Gaussian primitives per anchor to  $k = 10$  in all our experiments. We use GSplat [48] to render the Gaussian primitives. The key difference is that we extend the dimensionality of the Gaussian color attributes from 3 to  $N + 3$ , where  $N$  is the number of objects in the scene, defined when assigning object IDs. This makes the semantic rendering of Gaussians efficient. In our experiments, the loss weight  $\lambda_{\text{SSIM}}$  is set to 0.2. For the 3DGS version, we set the volume weight  $\lambda_{\text{vol}}$  to 0.0002 on the 3DOVS dataset, 0.00005 on the LERF-Mask dataset, and 0.00002 on the Replica and ScanNet datasets. For the 2DGS version, we reduce the  $\lambda_{\text{vol}}$  weight by half compared to the 3DGS version. We train each scene for 30,000 iterations on a single A800 GPU. In the case of the LERF-Mask dataset, we set  $\lambda_{\text{semantic}}$  to 0.01, while for other scenes, we set  $\lambda_{\text{semantic}}$  to 0.1.

### 4.2. Comparison with the State-of-the-arts

We provide more visualization results (Figs. 9 to 14) in the supplementary materials.

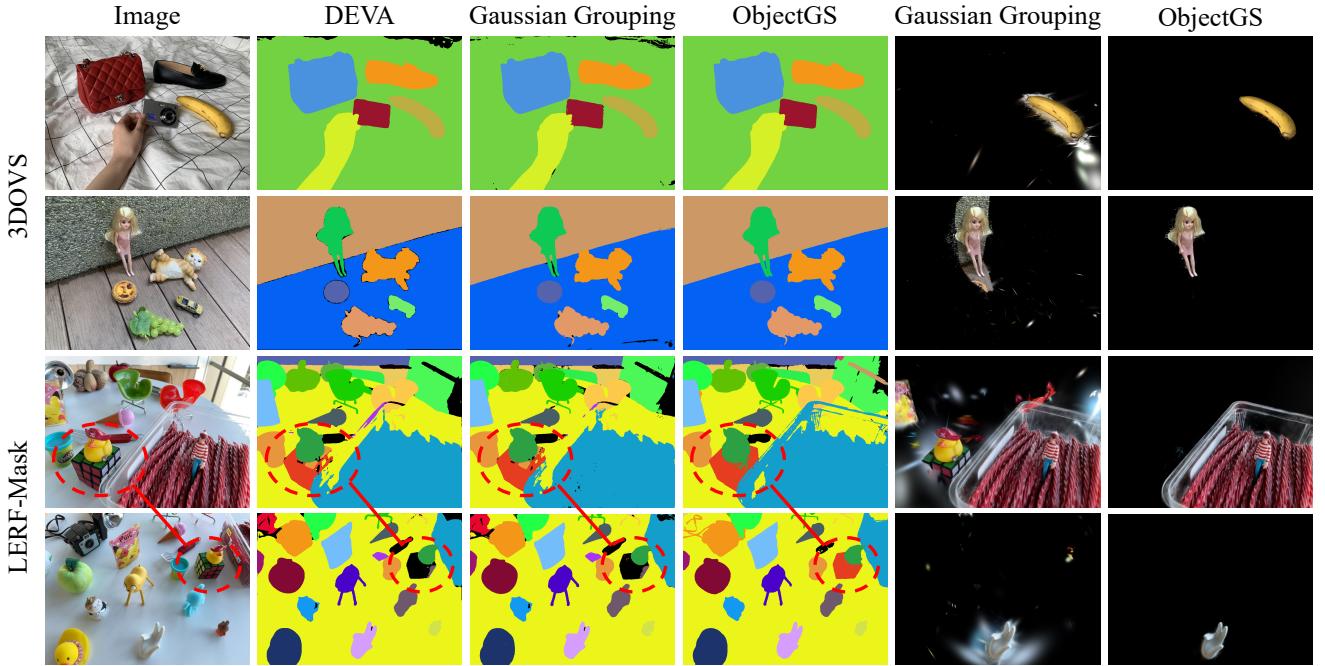


Figure 5. Qualitative comparison of open-vocabulary segmentation and 3D object queries. The red box highlights that our method can achieve multi-view consistent instance segmentation. In 3D object queries, our method has more accurate object segmentation boundaries.

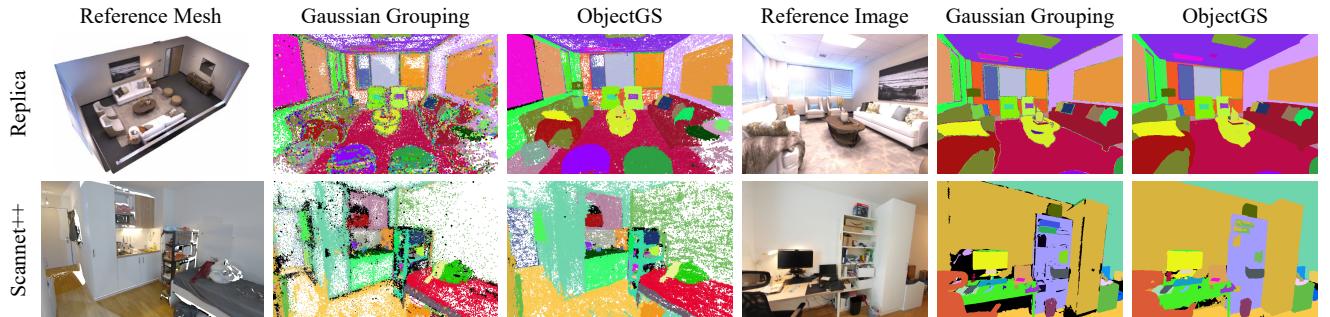


Figure 6. Qualitative comparison of panoptic segmentation. We visualize the segmentation of anchors (ours) and Gaussians (Gaussian Grouping [47]) using point clouds, where our results are more consistent and have less noise in 3D space. In 2D instance segmentation, our results have fewer holes and clearer boundaries.

Table 2. Panoptic segmentation results on Replica and ScanNet++ datasets. We randomly select 7 scenes in ScanNet++ for test.

Model	Dataset	PSNR	SSIM	LPIPS	IoU	Dice	Acc
Gaussian Grouping	Replica	39.52	0.9785	0.0548	83.36	91.84	94.70
ObjectGS(Ours)	Replica	<b>40.26</b>	<b>0.9842</b>	<b>0.0280</b>	<b>88.39</b>	<b>92.39</b>	<b>95.65</b>
Gaussian Grouping	ScanNet++	28.35	0.9296	0.1641	89.82	92.91	98.44
ObjectGS(Ours)	ScanNet++	<b>30.24</b>	<b>0.9327</b>	<b>0.1488</b>	<b>95.38</b>	<b>97.48</b>	<b>99.07</b>

Table 3. Comparison of 3D Instance Segmentation on ScanNet++

Method	Chamfer Distance ↓	Precision ↑	Recall ↑	F1 Score ↑
Gaussian Grouping	0.1472	35.9%	66.5%	41.6%
ObjectGS(Ours)	<b>0.1132</b>	<b>36.3%</b>	<b>86.1%</b>	<b>43.4%</b>

Table 4. Open-vocabulary segmentation results on 3DOVS dataset. We report IoU metric to compare with other methods.

Method	bed	bench	room	lawn	sofa	MEAN
LSEG [21]	56.0	6.0	19.2	4.5	17.5	20.6
OVSeg [22]	79.8	88.9	71.4	66.1	81.2	77.5
LERF [17]	73.5	53.2	46.6	27.0	73.7	54.8
3DOVS [24]	89.5	89.3	92.8	74.0	88.2	86.8
Langsplat [33]	77.8	77.3	58.4	90.9	60.2	73.0
Gaussian Grouping [47]	64.5	95.6	96.4	97.0	91.3	89.1
SAGA [4]	97.4	95.4	<b>96.8</b>	96.6	93.5	96.0
LBG [5]	97.7	96.3	95.9	<b>97.3</b>	87.4	94.9
ObjectGS(Ours)	<b>98.0</b>	<b>96.4</b>	95.1	97.2	<b>95.4</b>	<b>96.4</b>

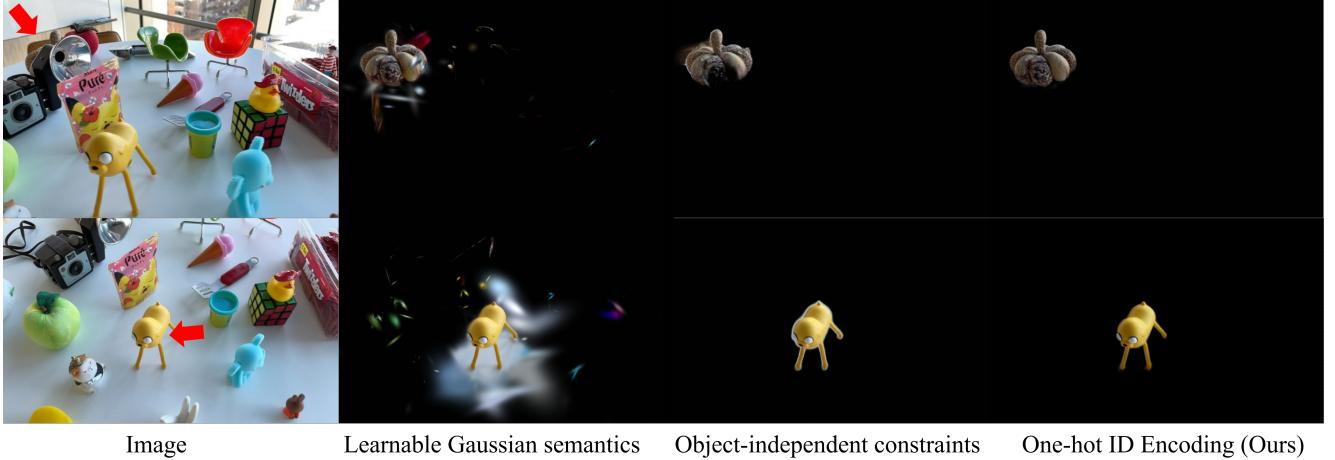


Figure 7. Qualitative comparison of different semantic modeling methods on 3D object query. Learnable Gaussian semantics leads to fuzzy positioning at the object boundary, and the constraint of object independence leads to ineffective object query under occlusion. In contrast, our proposed one-hot ID encoding overcomes both problems and achieves accurate 3D object query.

**Open-Vocabulary Segmentation (OVS).** Tabs. 1 and 4 show the performance when using text prompts to query objects from LERF-Mask and 3DOVS datasets. We use IoU (Intersection over Union) and Boundary IoU as our evaluation metrics. Our method significantly outperforms other approaches on both two OVS benchmarks, demonstrating the superiority of our unique framework design. We also provide a qualitative comparison in Figs. 5, 9 and 10 against state-of-the-art methods, where our approach fills most of the mask holes automatically and achieves more precise object segmentation. Notably, benefiting from the object id design bound to the anchor, our method can query the target object more accurately and conveniently than Gaussian grouping [47] without any post-processing. Besides, in addition to supporting text-based object queries, our method also supports click-based object queries, which is similar to the implementation in SAGA [4] and Click Gaussian [8].

**Panoptic Segmentation.** Tab. 2 demonstrates the performance when lifting the 2D object masks to 3D from Replica and ScanNet++ datasets. We use IoU, Dice coefficient, and Pixel Accuracy as our evaluation metrics. Experimental results show that our method outperforms Gaussian Grouping [47] in both reconstruction accuracy and segmentation precision. We provide visualizations of the segmentation results in Figs. 6, 13 and 14, demonstrating that our approach produces fewer holes and captures more accurate details. More importantly, we visualize the semantics of the point cloud derived from anchors and Gaussians to compare 3D instance segmentation performance. As shown in Figs. 6, 11 and 12, the point cloud produced by our method exhibit consistent semantics in 3D, whereas Gaussian Grouping [47] struggles to maintain this 3D semantic consistency. To validate the model performance in 3D Segmentation, we design an evaluation on ScanNet++

Table 5. Ablation of Gaussian semantic modeling on figurines scene of LERF-Mask dataset.

Setting	mIoU	mBIoU	PSNR	SSIM	LPIPS
Learnable Gaussian Semantics	69.57	67.86	25.67	0.8876	0.1584
Object-independent constraints	37.48	35.21	25.14	0.8911	0.1741
One-hot ID Encoding (Ours)	<b>88.19</b>	<b>85.22</b>	<b>26.75</b>	<b>0.9134</b>	<b>0.1386</b>

dataset: for each instance, we compute Chamfer Distance and F1 score between the reconstructed and ground-truth point clouds, counting a predicted point as a true positive if it lies within  $\tau=0.02\text{m}$  of any ground-truth point. As shown in Table 3, our model outperforms GaussianGrouping in all four metrics. We attribute this to our discrete Gaussian semantic modeling, which ensures that the semantics of different objects remain distinct and unaffected by one another.

### 4.3. Ablation Study

To comprehensively demonstrate the effectiveness of each component of our method, we design a series of ablation studies on the LERF-Mask and Replica datasets.

**Gaussian Semantic Modeling.** We conduct an ablation study on the figurines scene of the LERF-mask dataset to demonstrate the superiority of our unique semantic modeling approach. Specifically, we compare our method with other semantic modeling methods in Sec. 3.3. As shown in Tab. 5, our proposed One-hot ID Encoding method significantly outperforms both alternatives, highlighting the effectiveness of our approach. We also visualize the results of rendering individual target objects for each method, as shown in Fig. 7. Due to the ambiguity introduced by learnable Gaussian semantics, it struggles to accurately segment the boundaries of objects. Although object-independent constraints can accurately segment the boundaries of ob-

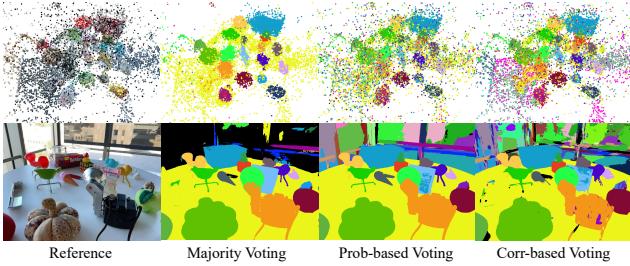


Figure 8. Ablation on different point cloud label initializations. The majority voting strategy is more robust in the foreground regions, while the probability-based and correspondence-based voting strategies show greater robustness in the background regions.

Table 6. Ablation on object ID voting strategy on figurines scene of LERF-Mask dataset.

	mIoU	mBIOU	PSNR	SSIM	LPIPS
Prob-based voting	84.46	81.46	25.69	0.9019	0.1586
Corr-based voting	59.67	57.50	26.13	0.9031	0.1539
Majority voting	<b>88.19</b>	<b>85.22</b>	<b>26.75</b>	<b>0.9134</b>	<b>0.1386</b>

Table 7. Ablation of Gaussian semantic loss weights on Replica.

$\lambda_{\text{semantic}}$	Acc	Dice	mIoU	PSNR	SSIM	LPIPS
0.00	0.00	0.00	0.00	40.19	0.9823	0.0288
0.01	94.75	90.70	86.15	<b>40.35</b>	0.9829	<b>0.0273</b>
0.10	<b>95.65</b>	<b>92.39</b>	<b>88.39</b>	40.26	<b>0.9842</b>	0.0280
1.00	94.42	90.98	86.67	35.43	0.9664	0.0866

jects, it is difficult to solve the rendering of objects in the case of occlusion. In contrast, our method combines the strengths of both approaches, enabling accurate object queries and robust scene decomposition simultaneously.

**Object ID Voting Strategy** Since the object ID prediction itself is prone to errors, lifting these predictions to the 3D point cloud inevitably introduces mislabeled points. To validate the robustness of our method, we design and compare three kinds of voting strategy to lift the object masks to 3D. As shown in Fig. 8 and Tab. 6, though the probability-based and correspondence-based strategy offer relatively more robust results in background regions, they produce suboptimal results when rendering foreground objects compared with the majority voting strategy. We argue that it is due to the grow-and-prune mechanism of our anchors, our method can naturally correct some of these mislabeled points over time. As a result, the simple majority voting strategy proves sufficient for most of the tested scenes.

**Gaussian Semantic loss.** To evaluate the effectiveness of semantic constraints, we test our method on the Replica dataset with different weights of semantic loss, as shown in Tab. 7. The results show that with a properly chosen loss weight, supervising Gaussian semantics helps improve both scene reconstruction and scene understanding.

#### 4.4. Application

Our explicit object-aware Gaussian representation enables several downstream applications post-training. We demonstrate two examples, as shown in our demo video:

**Object Mesh Extraction.** For object mesh extraction, we leverage our 2DGS-based variant. Specifically, we replace 3DGS primitives with 2DGS [12] because 2DGS typically better represents object surfaces. Once the scene is reconstructed, we can select target objects using either text prompts or click prompts. Since the object ID is directly bound to the anchor, we can use the anchors with the corresponding ID to generate the 2DGS model of the target object. We then apply TSDF Fusion, as suggested by 2DGS, to export the target object’s mesh.

**Scene Editing.** For scene editing, we adopt strategies similar to Gaussian Grouping [47]. Moreover, our method can more conveniently select the editing object, without calling the classifier. For example, object removal can be easily achieved by deleting the anchors associated with the target object’s ID. To recolor objects, we directly modify the color attributes of the associated Gaussians.

#### 5. Limitation

Although our method achieves robust open-world scene reconstruction and understanding in our test scenarios, some limitations still exist. Like existing approaches, we rely on 2D segmentation models [7, 37] to extract object masks. Therefore, when the segmentation model is unavailable or produces severely erroneous outputs, our method may fail. However, our approach is not merely a direct fitting of the 2D segmentation results. In our experimental results (*i.e.* Figs. 5 and 6), our method demonstrates fewer holes and more 3D-consistent results than the ground truth, indicating that our method can leverage scene geometry to infer unclassified semantics or correct misclassified semantics.

#### 6. Conclusion

We propose ObjectGS, an object-aware Gaussian splatting framework for open-world 3D scene reconstruction and 3D scene understanding. Unlike existing methods that distill Gaussian semantics, we optimize object-aware anchors to adjust Gaussian semantics. This design enables our method to perceive objects during reconstruction and adaptively build Gaussian representations based on the needs of individual objects. Furthermore, unlike existing approaches that optimize learnable Gaussian semantics, we model discrete Gaussian semantics and introduce a classification loss. This way ensures that Gaussians from different categories do not interfere during rendering. Finally, we demonstrate the extensibility of our method through its applications in object mesh extraction and scene editing, showcasing its versatility in downstream tasks.

## Acknowledgement

The work was supported by National Key Research and Development Program of China (2024YFB3909902), National Nature Science Foundation of China (62121002), Youth Innovation Promotion Association of CAS, and HKU Startup Fund.

## References

- [1] Jonathan T Barron, Ben Mildenhall, Matthew Tancik, Peter Hedman, Ricardo Martin-Brualla, and Pratul P Srinivasan. Mip-nerf: A multiscale representation for anti-aliasing neural radiance fields. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 5855–5864, 2021. 1
- [2] Mathilde Caron, Hugo Touvron, Ishan Misra, Hervé Jégou, Julien Mairal, Piotr Bojanowski, and Armand Joulin. Emerging properties in self-supervised vision transformers. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2021. 2
- [3] Jiazhong Cen, Zanwei Zhou, Jiemin Fang, Wei Shen, Lingxi Xie, Dongsheng Jiang, Xiaopeng Zhang, Qi Tian, et al. Segment anything in 3d with nerfs. *Proceedings of the International Conference on Neural Information Processing Systems*, 36:25971–25990, 2023. 2, 5
- [4] Jiazhong Cen, Jiemin Fang, Chen Yang, Lingxi Xie, Xiaopeng Zhang, Wei Shen, and Qi Tian. Segment any 3d gaussians. In *Proceedings of the AAAI Conference on Artificial Intelligence*, pages 1971–1979, 2025. 1, 2, 4, 6, 7
- [5] Rohan Chacko, Nicolai Haeni, Eldar Khaliullin, Lin Sun, and Douglas Lee. Lifting by gaussians: A simple, fast and flexible method for 3d instance segmentation. *arXiv preprint arXiv:2502.00173*, 2025. 4, 6
- [6] Bowen Cheng, Ishan Misra, Alexander G Schwing, Alexander Kirillov, and Rohit Girdhar. Masked-attention mask transformer for universal image segmentation. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 1290–1299, 2022. 2
- [7] Ho Kei Cheng, Seoung Wug Oh, Brian Price, Alexander Schwing, and Joon-Young Lee. Tracking anything with decoupled video segmentation. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 1316–1326, 2023. 2, 3, 5, 8
- [8] Seokhun Choi, Hyeonseop Song, Jaechul Kim, Taehyeong Kim, and Hoseok Do. Click-gaussian: Interactive segmentation to any 3d gaussians. In *European Conference on Computer Vision*, pages 289–305. Springer, 2024. 1, 2, 7
- [9] Rahul Goel, Dhawal Sirikonda, Saurabh Saini, and PJ Narayanan. Interactive segmentation of radiance fields. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 4201–4211, 2023. 2
- [10] Haoyu Guo, Sida Peng, Haotong Lin, Qianqian Wang, Guofeng Zhang, Hujun Bao, and Xiaowei Zhou. Neural 3d scene reconstruction with the manhattan-world assumption. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 5511–5520, 2022. 5
- [11] Haoyu Guo, He Zhu, Sida Peng, Yuang Wang, Yujun Shen, Ruizhen Hu, and Xiaowei Zhou. Sam-guided graph cut for 3d instance segmentation. In *European Conference on Computer Vision*, pages 234–251. Springer, 2024. 3
- [12] Binbin Huang, Zehao Yu, Anpei Chen, Andreas Geiger, and Shenghua Gao. 2d gaussian splatting for geometrically accurate radiance fields. In *ACM SIGGRAPH Conference*, pages 1–11, 2024. 2, 8
- [13] Youngkyoon Jang, Jiali Zheng, Jifei Song, Helisa Dhamo, Eduardo Pérez-Pellitero, Thomas Tanay, Matteo Maggioni, Richard Shaw, Sibi Catley-Chandar, Yiren Zhou, et al. Vschh 2023: A benchmark for the view synthesis challenge of human heads. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 1121–1128, 2023. 2
- [14] Lihuan Jiang, Yucheng Mao, Lining Xu, Tao Lu, Kerui Ren, Yichen Jin, Xudong Xu, Mulin Yu, Jiangmiao Pang, Feng Zhao, et al. Anysplat: Feed-forward 3d gaussian splatting from unconstrained views. *arXiv preprint arXiv:2505.23716*, 2025. 2
- [15] Lihuan Jiang, Kerui Ren, Mulin Yu, Lining Xu, Junting Dong, Tao Lu, Feng Zhao, Dahua Lin, and Bo Dai. Horizons: Unified 3d gaussian splatting for large-scale aerial-to-ground scenes. In *Proceedings of the Computer Vision and Pattern Recognition Conference*, pages 26789–26799, 2025. 2
- [16] Bernhard Kerbl, Georgios Kopanas, Thomas Leimkühler, and George Drettakis. 3d gaussian splatting for real-time radiance field rendering. *ACM Transactions on Graphics*, 42(4):139–1, 2023. 1, 2
- [17] Justin Kerr, Chung Min Kim, Ken Goldberg, Angjoo Kanazawa, and Matthew Tancik. Lerf: Language embedded radiance fields. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 19729–19739, 2023. 5, 6
- [18] 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 *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 4015–4026, 2023. 1, 2
- [19] Sosuke Kobayashi, Eiichi Matsumoto, and Vincent Sitzmann. Decomposing nerf for editing via feature field distillation. *Proceedings of the International Conference on Neural Information Processing Systems*, 35:23311–23330, 2022. 2
- [20] Lingdong Kong, Yaru Niu, Shaoyuan Xie, Hanjiang Hu, Lai Xing Ng, Benoit R Cottreau, Liangjun Zhang, Hesheng Wang, Wei Tsang Ooi, Ruijie Zhu, et al. The robodepth challenge: Methods and advancements towards robust depth estimation. *arXiv preprint arXiv:2307.15061*, 2023. 2
- [21] Boyi Li, Kilian Q Weinberger, Serge Belongie, Vladlen Koltun, and René Ranftl. Language-driven semantic segmentation. *arXiv preprint arXiv:2201.03546*, 2022. 6
- [22] Feng Liang, Bichen Wu, Xiaoliang Dai, Kunpeng Li, Yinan Zhao, Hang Zhang, Peizhao Zhang, Peter Vajda, and Diana Marculescu. Open-vocabulary semantic segmentation with mask-adapted clip. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 7061–7070, 2023. 6

- [23] Siyun Liang, Sen Wang, Kunyi Li, Michael Niemeyer, Stefano Gasperini, Nassir Navab, and Federico Tombari. Superseg: Open-vocabulary 3d segmentation with structured super-gaussians. *arXiv preprint arXiv:2412.10231*, 2024. 1, 2, 4
- [24] Kunhao Liu, Fangneng Zhan, Jiahui Zhang, Muyu Xu, Yingchen Yu, Abdulmotaleb El Saddik, Christian Theobalt, Eric Xing, and Shijian Lu. Weakly supervised 3d open-vocabulary segmentation. *Advances in Neural Information Processing Systems*, 36:53433–53456, 2023. 5, 6
- [25] Li Liu, Ruijie Zhu, Jiacheng Deng, Ziyang Song, Wenfei Yang, and Tianzhu Zhang. Plane2depth: Hierarchical adaptive plane guidance for monocular depth estimation. *IEEE Transactions on Circuits and Systems for Video Technology*, 2024. 2
- [26] Jiahao Lu, Jiacheng Deng, Ruijie Zhu, Yanzhe Liang, Wenfei Yang, Xu Zhou, and Tianzhu Zhang. Dn-4dgs: Denoised deformable network with temporal-spatial aggregation for dynamic scene rendering. *Advances in Neural Information Processing Systems*, 37:84114–84138, 2024. 2
- [27] Tao Lu, Mulin Yu, Linning Xu, Yuanbo Xiangli, Limin Wang, Dahua Lin, and Bo Dai. Scaffold-gs: Structured 3d gaussians for view-adaptive rendering. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 20654–20664, 2024. 1, 2, 3, 5
- [28] Weijie Lyu, Xuetong Li, Abhijit Kundu, Yi-Hsuan Tsai, and Ming-Hsuan Yang. Gaga: Group any gaussians via 3d-aware memory bank. *arXiv preprint arXiv:2404.07977*, 2024. 2, 5
- [29] Ben Mildenhall, Pratul P Srinivasan, Matthew Tancik, Jonathan T Barron, Ravi Ramamoorthi, and Ren Ng. NeRF: Representing scenes as neural radiance fields for view synthesis. In *Proceedings of the European Conference on Computer Vision*, pages 405–421, 2020. 1, 2
- [30] Ashkan Mirzaei, Tristan Amentado-Armstrong, Konstantinos G Derpanis, Jonathan Kelly, Marcus A Brubaker, Igor Gilitschenski, and Alex Levinshtein. Spin-nerf: Multiview segmentation and perceptual inpainting with neural radiance fields. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 20669–20679, 2023. 2
- [31] Maxime Oquab, Timothée Darcet, Théo Moutakanni, Huy Vo, Marc Szafraniec, Vasil Khalidov, Pierre Fernandez, Daniel Haziza, Francisco Massa, Alaeldin El-Nouby, et al. Dinov2: Learning robust visual features without supervision. *arXiv preprint arXiv:2304.07193*, 2023. 2
- [32] Yuning Peng, Haiping Wang, Yuan Liu, Chenglu Wen, Zhen Dong, and Bisheng Yang. Gags: Granularity-aware feature distillation for language gaussian splatting. *arXiv preprint arXiv:2412.13654*, 2024. 2, 4
- [33] Minghan Qin, Wanhua Li, Jiawei Zhou, Haoqian Wang, and Hanspeter Pfister. Langsplat: 3d language gaussian splatting. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 20051–20060, 2024. 1, 2, 4, 5, 6
- [34] Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, et al. Learn-ing transferable visual models from natural language super- vision. In *Proceedings of the International Conference on Machine Learning*, pages 8748–8763. PMLR, 2021. 2
- [35] Nikhila Ravi, Valentin Gabeur, Yuan-Ting Hu, Ronghang Hu, Chaitanya Ryali, Tengyu Ma, Haitham Khedr, Roman Rädle, Chloe Rolland, Laura Gustafson, et al. Sam 2: Segment anything in images and videos. *arXiv preprint arXiv:2408.00714*, 2024. 2
- [36] Kerui Ren, Lihan Jiang, Tao Lu, Mulin Yu, Linning Xu, Zhangkai Ni, and Bo Dai. Octree-gs: Towards consistent real-time rendering with lod-structured 3d gaussians. *arXiv preprint arXiv:2403.17898*, 2024. 2
- [37] 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, 3, 4, 8
- [38] Ziyang Song, Zerong Wang, Bo Li, Hao Zhang, Ruijie Zhu, Li Liu, Peng-Tao Jiang, and Tianzhu Zhang. Depthmaster: Taming diffusion models for monocular depth estimation. *arXiv preprint arXiv:2501.02576*, 2025. 2
- [39] Julian Straub, Thomas Whelan, Lingni Ma, Yufan Chen, Erik Wijmans, Simon Green, Jakob J Engel, Raul Mur-Artal, Carl Ren, Shobhit Verma, et al. The replica dataset: A digital replica of indoor spaces. *arXiv preprint arXiv:1906.05797*, 2019. 5
- [40] Ayça Takmaz, Elisabetta Fedele, Robert W Sumner, Marc Pollefeys, Federico Tombari, and Francis Engelmann. Open-mask3d: open-vocabulary 3d instance segmentation. In *Proceedings of the International Conference on Neural Information Processing Systems*, pages 68367–68390, 2023. 2
- [41] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. *Advances in neural information processing systems*, 30, 2017. 2
- [42] Dor Verbin, Peter Hedman, Ben Mildenhall, Todd Zickler, Jonathan T Barron, and Pratul P Srinivasan. Ref-nerf: Structured view-dependent appearance for neural radiance fields. In *2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 5481–5490. IEEE, 2022. 1
- [43] Qianyi Wu, Xian Liu, Yuedong Chen, Kejie Li, Chuanxia Zheng, Jianfei Cai, and Jianmin Zheng. Object-compositional neural implicit surfaces. In *European Conference on Computer Vision*, pages 197–213. Springer, 2022. 5
- [44] Qianyi Wu, Kaisiyuan Wang, Kejie Li, Jianmin Zheng, and Jianfei Cai. Objectsdf++: Improved object-compositional neural implicit surfaces. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 21764–21774, 2023. 4
- [45] Yanmin Wu, Jiarui Meng, Hajjie Li, Chenming Wu, Yahao Shi, Xinhua Cheng, Chen Zhao, Haocheng Feng, Errui Ding, Jingdong Wang, et al. Opengaussian: Towards point-level 3d gaussian-based open vocabulary understanding. *arXiv preprint arXiv:2406.02058*, 2024. 1
- [46] Yunhan Yang, Xiaoyang Wu, Tong He, Hengshuang Zhao, and Xihui Liu. Sam3d: Segment anything in 3d scenes. *arXiv preprint arXiv:2306.03908*, 2023. 2

- [47] Mingqiao Ye, Martin Danelljan, Fisher Yu, and Lei Ke. Gaussian grouping: Segment and edit anything in 3d scenes. In *Proceedings of the European Conference on Computer Vision*, pages 162–179, 2024. [1](#), [2](#), [3](#), [5](#), [6](#), [7](#), [8](#)
- [48] Vickie Ye, Rui long Li, Justin Kerr, Matias Turkulainen, Brent Yi, Zhuoyang Pan, Otto Seiskari, Jianbo Ye, Jeffrey Hu, Matthew Tancik, et al. gsplat: An open-source library for gaussian splatting. *Journal of Machine Learning Research*, 26(34):1–17, 2025. [1](#), [5](#)
- [49] Chandan Yeshwanth, Yueh-Cheng Liu, Matthias Nießner, and Angela Dai. Scannet++: A high-fidelity dataset of 3d indoor scenes. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 12–22, 2023. [5](#)
- [50] Haiyang Ying, Yixuan Yin, Jinzhi Zhang, Fan Wang, Tao Yu, Ruqi Huang, and Lu Fang. Omniseg3d: Omnipractical 3d segmentation via hierarchical contrastive learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 20612–20622, 2024. [2](#)
- [51] Mulin Yu, Tao Lu, Linning Xu, Lihan Jiang, Yuanbo Xiangli, and Bo Dai. Gsdf: 3dgs meets sdf for improved neural rendering and reconstruction. *Advances in Neural Information Processing Systems*, 37:129507–129530, 2024. [2](#)
- [52] Zehao Yu, Anpei Chen, Binbin Huang, Torsten Sattler, and Andreas Geiger. Mip-splatting: Alias-free 3d gaussian splatting. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 19447–19456, 2024. [1](#)
- [53] Pierluigi Zama Ramirez, Alex Costanzino, Fabio Tosi, Matteo Poggi, Luigi Di Stefano, Jean-Baptiste Weibel, Dominik Bauer, Doris Antensteiner, Markus Vincze, Jiaqi Li, et al. Tricky 2024 challenge on monocular depth from images of specular and transparent surfaces. In *European Conference on Computer Vision*, pages 248–266. Springer, 2024. [2](#)
- [54] Ruijie Zhu, Jiahao Chang, Ziyang Song, Jiahuan Yu, and Tianzhu Zhang. Tiface: Improving facial reconstruction through tensorial radiance fields and implicit surfaces. *arXiv preprint arXiv:2312.09527*, 2023.
- [55] Ruijie Zhu, Ziyang Song, Li Liu, Jianfeng He, Tianzhu Zhang, and Yongdong Zhang. Ha-bins: Hierarchical adaptive bins for robust monocular depth estimation across multiple datasets. *IEEE Transactions on Circuits and Systems for Video Technology*, 34(6):4354–4366, 2023. [2](#)
- [56] Ruijie Zhu, Yanzhe Liang, Hanzhi Chang, Jiacheng Deng, Jiahao Lu, Wenfei Yang, Tianzhu Zhang, and Yongdong Zhang. Motions: Exploring explicit motion guidance for deformable 3d gaussian splatting. *Advances in Neural Information Processing Systems*, 37:101790–101817, 2024. [2](#)
- [57] Ruijie Zhu, Chuxin Wang, Ziyang Song, Li Liu, Tianzhu Zhang, and Yongdong Zhang. Scaleddepth: Decomposing metric depth estimation into scale prediction and relative depth estimation. *arXiv preprint arXiv:2407.08187*, 2024. [2](#)

# ObjectGS: Object-aware Scene Reconstruction and Scene Understanding via Gaussian Splatting

## Supplementary Material

### 7. Training Overhead

Table 8 compares training time, FPS, and GPU memory across different instance counts. Even with about 100 instances, overhead remains minimal with efficient parallel rasterizer. Notably, since our one-hot ID encoding is not learnable parameters, it will not significantly increase training overhead. Meanwhile, we can optionally encode only a subset of target instances or leverage category hierarchies, avoiding the waste and inflexibility of fixed-length representations under long-tailed distributions. Therefore, in real applications, our method is both more flexible and scalable.

### 8. Voting Algorithm

We provide the pseudo code of Algorithms 1 to 3 to clearly demonstrate the proposed voting strategies.

### 9. More Visualization

We provide more visualization results as shown in Figs. 9 to 14, which includes visualization of OVS segmentation results, panoptic segmentation results, and 3D instance segmentation with point clouds.

---

#### Algorithm 1 Object ID Majority Voting

---

```

1: Input:
2: Point cloud:  $P_{3D} = \{p_1, p_2, \dots, p_M\}$ 
3: Object ID maps:  $L = \{L_1, L_2, \dots, L_N\}$ 
4: Camera poses:  $C = \{C_1, C_2, \dots, C_N\}$ 
5: Initialization:
6: labels =  $\emptyset$ 
7: for each point  $p_i \in P_{3D}$  do
8:   for each camera pose  $C_j \in C$  do
9:      $x_i = \text{Project}(p_i, C_j)$ 
10:    Append  $L_j(x_i)$  to labels[ $p_i$ ]
11:   end for
12: end for
13: for each point  $p_i \in P_{3D}$  do
14:   if labels[ $p_i$ ]  $\neq \emptyset$  then
15:     frequency(ID) = Counter(labels[ $p_i$ ])
16:     ID = arg max frequency(ID)
17:   end if
18:   Update  $p_i = (x_i, y_i, z_i, \text{object ID})$ 
19: end for
20: Output: Updated point cloud  $P_{3D}$  with object IDs.

```

---



---

#### Algorithm 2 Object ID Probability-based Voting

---

```

1: Input:
2: Point cloud:  $P_{3D} = \{p_1, p_2, \dots, p_M\}$ 
3: Object ID maps:  $L = \{L_1, L_2, \dots, L_N\}$ 
4: Camera poses:  $C = \{C_1, C_2, \dots, C_N\}$ 
5: Initialization:
6: labels =  $\emptyset$ 
7: for each point  $p_i \in P_{3D}$  do
8:   for each camera pose  $C_j \in C$  do
9:      $x_i = \text{Project}(p_i, C_j)$ 
10:    Append  $L_j(x_i)$  to labels[ $p_i$ ]
11:   end for
12: end for
13: for each point  $p_i \in P_{3D}$  do
14:   if labels[ $p_i$ ]  $\neq \emptyset$  then
15:     frequency(ID) = Counter(labels[ $p_i$ ])
16:     ID = Random(Prob = frequency(ID))
17:   end if
18:   Update  $p_i = (x_i, y_i, z_i, \text{object ID})$ 
19: end for
20: Output: Updated point cloud  $P_{3D}$  with object IDs.

```

---



---

#### Algorithm 3 Object ID Correspondence-based Voting

---

```

1: Input:
2: Point cloud:  $P_{3D} = \{p_1, p_2, \dots, p_M\}$ 
3: Object ID maps:  $L = \{L_1, L_2, \dots, L_N\}$ 
4: Correspondences:  $C = \{C_1, C_2, \dots, C_N\}$ 
5: Initialization:
6: labels =  $\emptyset$ 
7: for each point  $p_i \in P_{3D}$  do
8:   for each correspondence  $C_j \in C$  do
9:      $x_i = \text{Project}(p_i, C_j)$ 
10:    Append  $L_j(x_i)$  to labels[ $p_i$ ]
11:   end for
12: end for
13: for each point  $p_i \in P_{3D}$  do
14:   if labels[ $p_i$ ]  $\neq \emptyset$  then
15:     frequency(ID) = Counter(labels[ $p_i$ ])
16:     ID = arg max frequency(ID)
17:   end if
18:   Update  $p_i = (x_i, y_i, z_i, \text{object ID})$ 
19: end for
20: Output: Updated point cloud  $P_{3D}$  with object IDs.

```

---

Table 8. Training time, FPS, and GPU memory comparison

Scene	#Objects	Training time		FPS		GPU memory	
		GS Grouping	Ours	GS Grouping	Ours	GS Grouping	Ours
bed (3DOVS)	7	94 min	72 min	100	80	~15G	~10G
sofa (3DOVS))	24	55 min	31 min	110	90	~18G	~12G
1ada (ScanNet++)	63	68 min	69 min	90	50	~40G	~35G
3e8b (ScanNet++)	80	71 min	113 min	80	40	~40G	~45G
0d2e (ScanNet++)	90	73 min	112 min	80	40	~40G	~45G

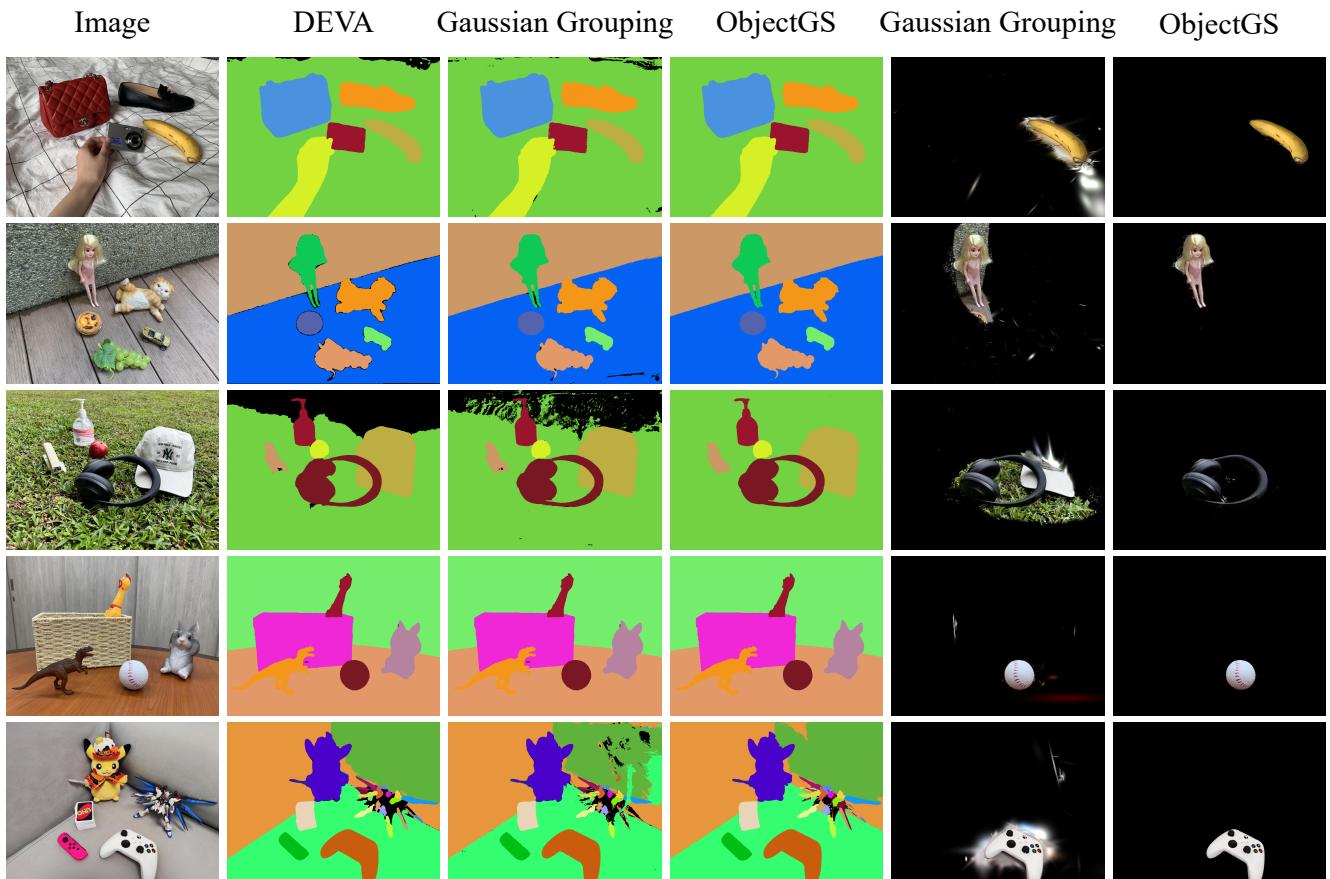
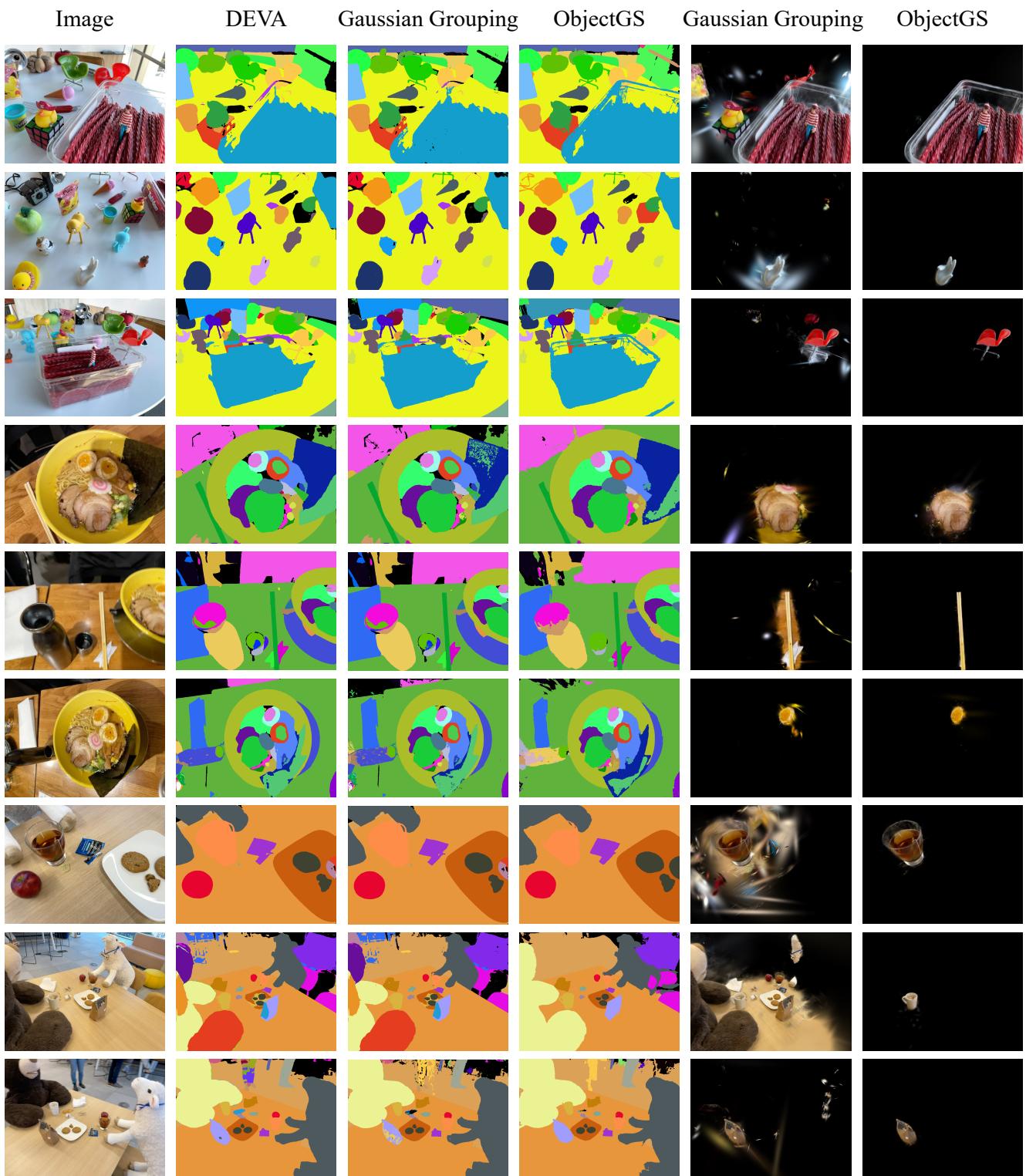


Figure 9. Qualitative comparison of open vocabulary segmentation and 3D object query on the 3DOVS dataset.



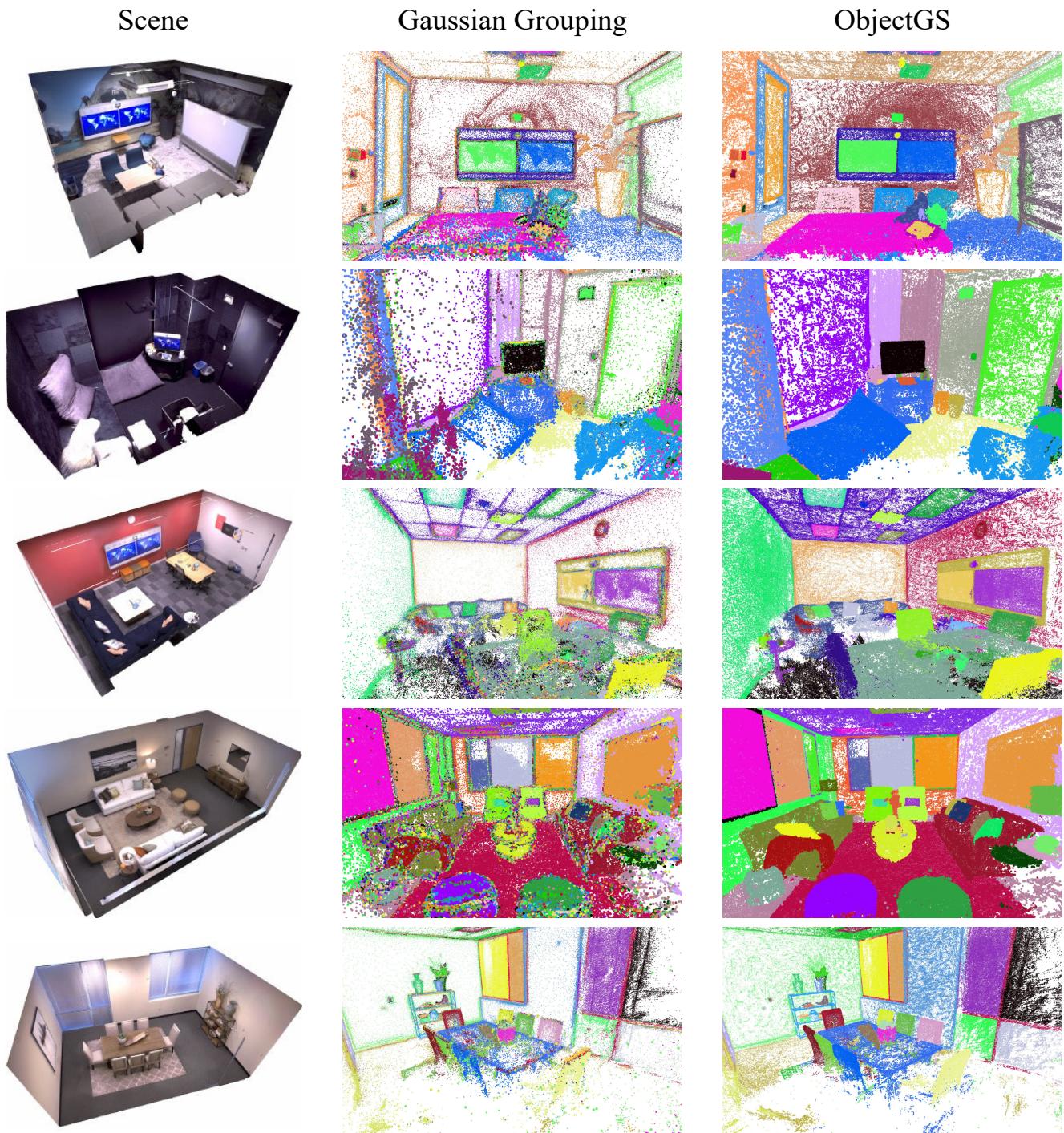


Figure 11. Qualitative comparison of 3D panoptic segmentation on the Replica dataset.

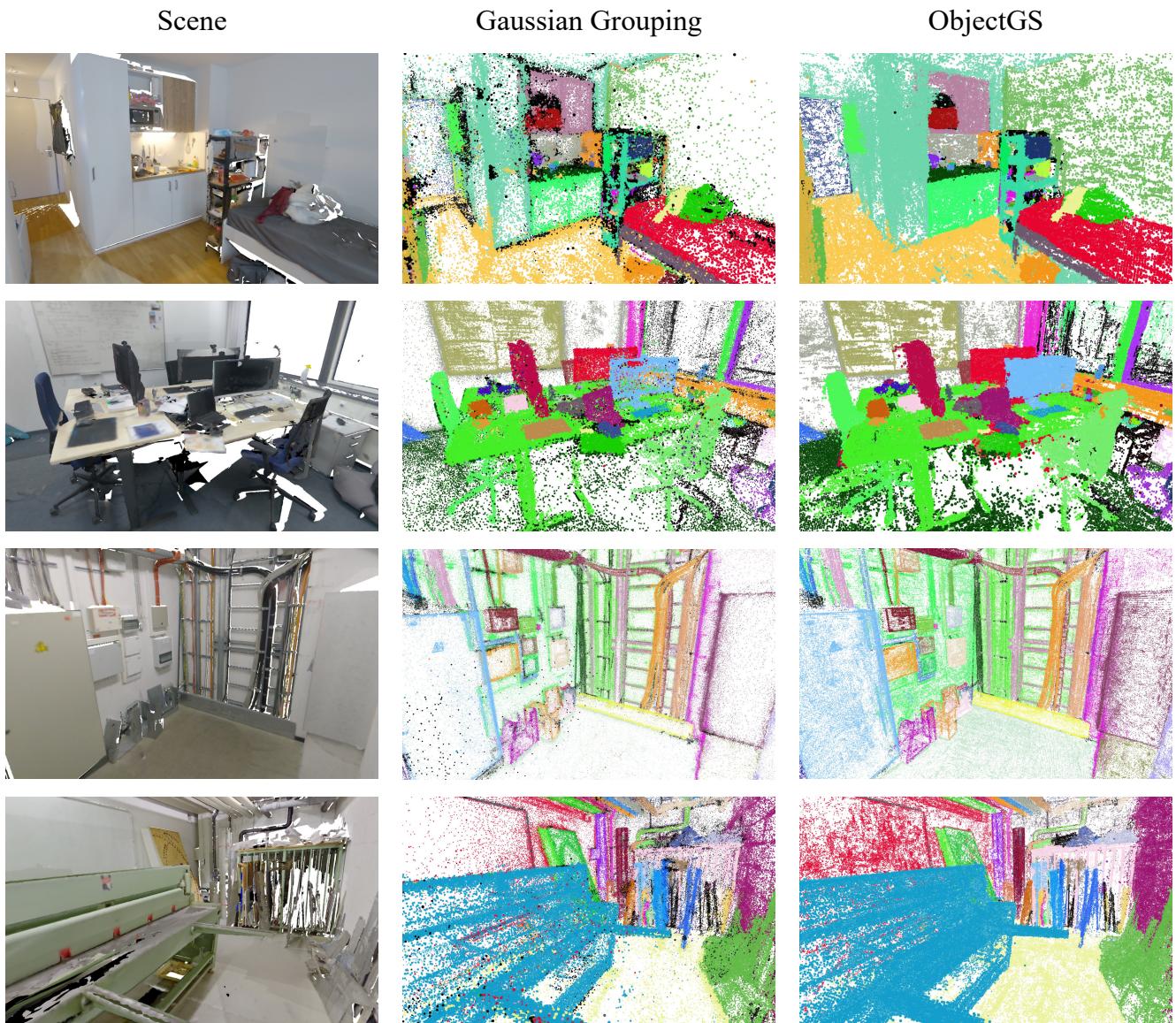


Figure 12. Qualitative comparison of 3D panoptic segmentation on the Scannet++ dataset.

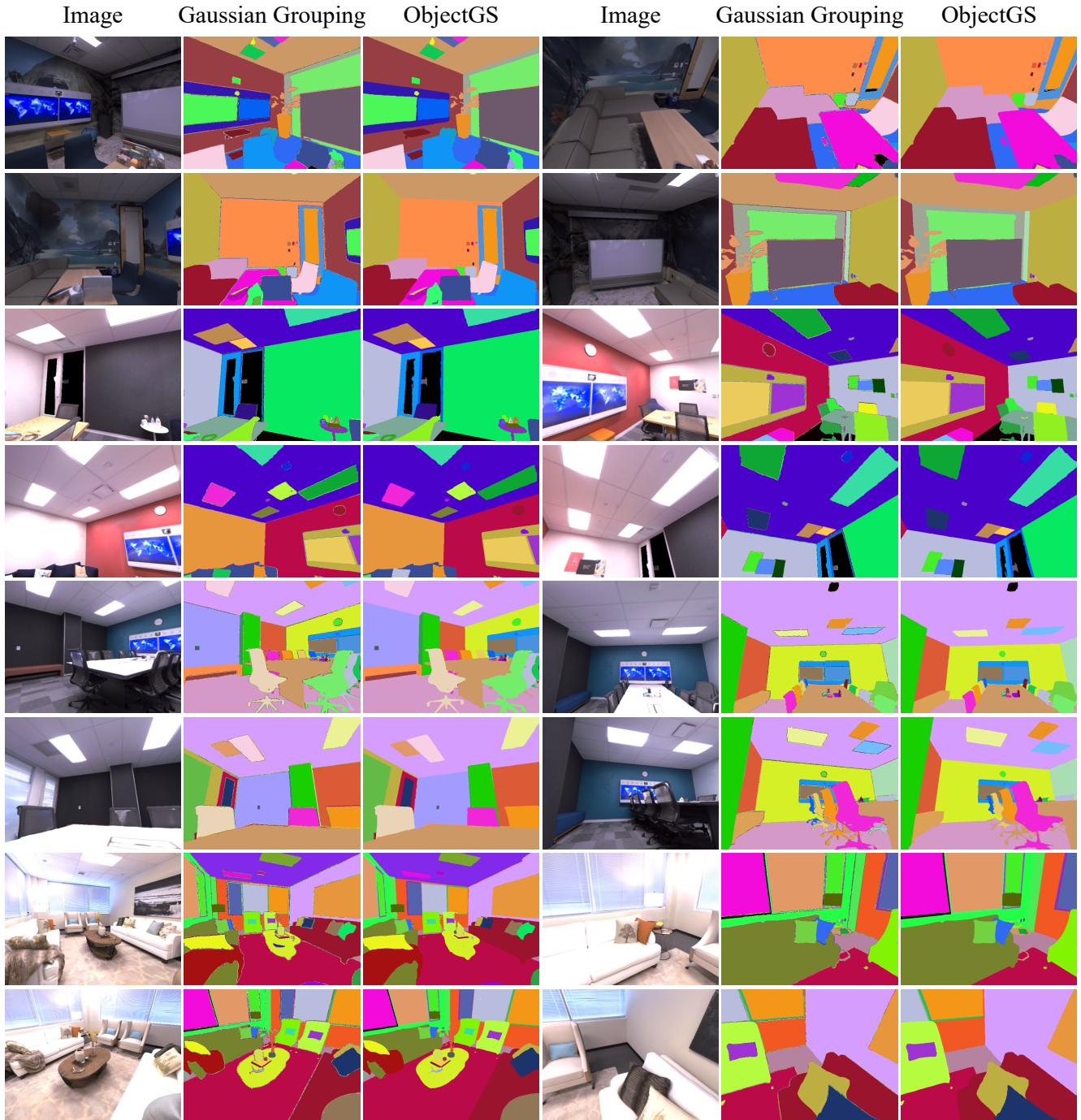


Figure 13. Qualitative comparison of 2D panoptic segmentation on the Replica dataset.

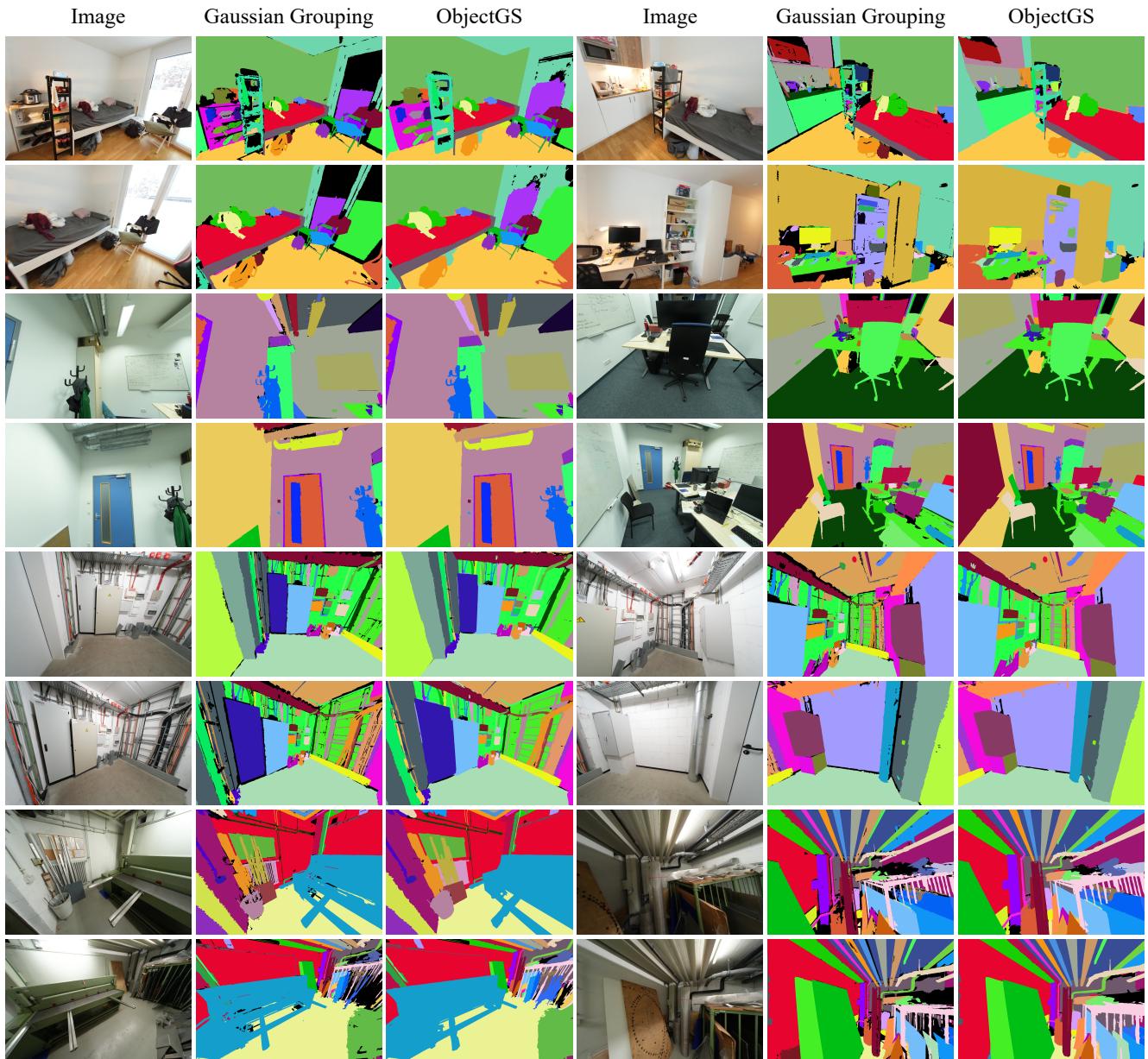


Figure 14. Qualitative comparison of 2D panoptic segmentation on the Scannet++ dataset.