

# Pointmap Association and Piecewise-Plane Constraint for Consistent and Compact 3D Gaussian Segmentation Field

Wenhai Hu<sup>1,2</sup>, Wenhao Chai<sup>3</sup>, Shengyu Hao<sup>1,2</sup>, Xiaotong Cui<sup>1</sup>, Xuexiang Wen<sup>1</sup>, Jenq-Neng Hwang<sup>3</sup>, Gaoang Wang<sup>1,2\*</sup>

<sup>1</sup>ZJU-UIUC Institute, Zhejiang University, 718 East Haizhou Rd., Haining, 314400, Zhejiang, China.

<sup>2</sup>College of Computer Science and Technology, Zhejiang University, 38 Zheda Rd., Hangzhou, 310027, Zhejiang, China.

<sup>3</sup>Department of Electrical & Computer Engineering, University of Washington, 185 Stevens Way, Seattle, 98195, Washington, USA.

\*Corresponding author(s). E-mail(s): [gaoangwang@intl.zju.edu.cn](mailto:gaoangwang@intl.zju.edu.cn);

Contributing authors: [whu@zju.edu.cn](mailto:whu@zju.edu.cn); [wchai@uw.edu](mailto:wchai@uw.edu); [shengyuahao@zju.edu.cn](mailto:shengyuahao@zju.edu.cn); [xiaotong.21@intl.zju.edu.cn](mailto:xiaotong.21@intl.zju.edu.cn); [xuexiangwen@zju.edu.cn](mailto:xuexiangwen@zju.edu.cn); [hwang@uw.edu](mailto:hwang@uw.edu);

## Abstract

Achieving a consistent and compact 3D segmentation field is crucial for maintaining semantic coherence across views and accurately representing scene structures. Previous 3D scene segmentation methods rely on video segmentation models to address inconsistencies across views, but the absence of spatial information often leads to object misassociation when object temporarily disappear and reappear. Furthermore, in the process of 3D scene reconstruction, segmentation and optimization are often treated as separate tasks. As a result, optimization typically lacks awareness of semantic category information, which can result in floaters with ambiguous segmentation. To address these challenges, we introduce CCGS, a method designed to achieve both view Consistent 2D segmentation and a Compact 3D Gaussian Segmentation field. CCGS incorporates pointmap association and a piecewise-plane constraint. First, we establish pixel correspondence between adjacent images by minimizing the Euclidean distance between their pointmaps. We then redefine object mask overlap accordingly. The Hungarian algorithm is employed to optimize mask association by minimizing the total matching cost, while allowing for partial matches. To further enhance compactness, the piecewise-plane constraint restricts point displacement within local planes during optimization, thereby preserving structural integrity. Experimental results on ScanNet and Replica datasets demonstrate that CCGS outperforms existing methods in both 2D panoptic segmentation and 3D Gaussian segmentation. [Project Page](#)

**Keywords:** View-consistent segmentation, 3D Gaussian segmentation field, Pointmap association

## 1 Introduction

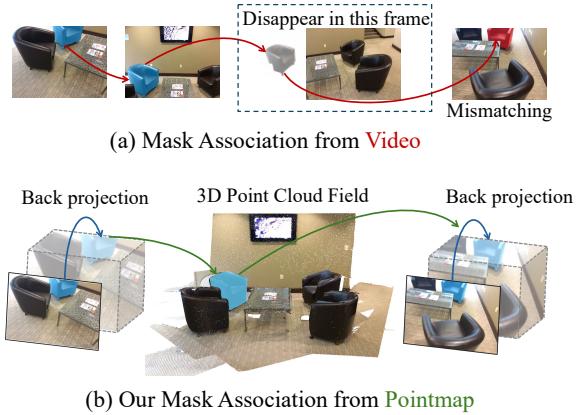
3D scene segmentation plays a crucial role in enhancing scene perception, understanding, and

interaction [43, 14, 30, 3, 55]. It facilitates diverse and advanced applications such as autonomous driving, augmented reality, and robotics by

enabling efficient navigation, precise object recognition, and intelligent interaction within complex environments [17, 23, 22, 54, 2]. Recent advancements have extensively explored NeRF-based segmentation methods [1, 46, 59, 49], which allow for segmenting and rendering of images from novel viewpoints. However, these methods often suffer from slow training and inference times, limiting their practicality in real-world scenarios. The emergence of Gaussian splatting [18] introduces a innovative approach that enables real-time 2D rendering while maintaining distinct spatial meaning for each Gaussian point. This advancement offers a new perspective on constructing 3D segmentation fields, addressing limitations of earlier methods.

Previous research [9, 13, 53, 56, 60, 57] highlights two main challenges in establishing a 3D Gaussian segmentation field. First, some existing methods [9, 56] rely on video segmentation models [6] for mask association, which lack spatial consistency. This limitation becomes evident in complex scenes where objects intermittently disappear and reappear in the image sequence due to occlusion or moving out of view. In such cases, the same object may be assigned different IDs, reducing the effectiveness of these methods. For instance, as shown in Figure 1, when an object like a chair exits the frame and reappears several frames later, the video segmentation model may mistakenly assign it a new ID, treating it as a different object. Second, segmentation and reconstruction are typically handled as separate tasks, leading to an optimization process that lacks semantic awareness and fails to incorporate meaningful structural information. The absence of effective semantic constraints during the optimization process often leads to the generation of meaningless floaters in empty space, which do not contribute to the segmentation field. Furthermore, the lack of constraints during the densification process allows replicated points to spread without restriction, resulting in ambiguous class boundaries. These limitations compromise both the accuracy and compactness of the 3D Gaussian segmentation field, making it less effective in representing complex scenes.

To address the aforementioned issues, we propose CCGS, a method combining pointmap-based association and a piecewise-plane constraint to



**Fig. 1** Differences in mask association: Video vs. Pointmap. Video segmentation often struggle to maintain consistency during significant changes in camera views. In contrast, constructing a unified 3D point cloud field can ensure segmentation accuracy by leveraging spatial information.

construct a consistent and compact 3D Gaussian segmentation field. This approach ensures the consistency of 2D segmentation while maintaining the compactness of the 3D segmentation. Specifically, we first determine the pixel correspondence between adjacent images by minimizing the Euclidean distance between their pointmaps. The overlap between object masks is redefined based on this correspondence, and a matching cost is computed to associate masks across frames. The mask association problem is solved using the Hungarian algorithm, optimizing the total matching cost while allowing partial matches. To achieve a compact 3D segmentation field, we introduce plane constrained Gaussian Splatting, where each point is restricted to a piecewise-plane defined by its nearest neighbors of the same class. This constraint enhances the compactness of the segmentation field during optimization. Additionally, in the densification process, we propose a split projection method that ensures replicated points remain within the neighborhood plane of similar points, preventing boundary blending between different classes and preserving segmentation accuracy.

In summary, our work makes the following contributions:

- We propose CCGS, a Gaussian segmentation field that achieves both view-consistent 2D segmentation and compact 3D segmentation.

- We propose pointmap association to generate unified 3D field that facilitates consistent 2D segmentation.
- We define piecewise-plane constrained Gaussian splatting, which restricts point displacement during optimization and densification to achieve a compact 3D segmentation.
- We conducted extensive experiments on multiple datasets, demonstrating that our approach achieves state-of-the-art (SOTA) performance in both 2D and 3D segmentation tasks.

## 2 Related Works

### 2.1 Point Cloud Segmentation

3D point cloud segmentation classifies the point cloud into meaningful regions or segments that belong to the same class [48, 34, 36]. Some 3D point cloud segmentation methods primarily rely on training closed-set models. PointNet [38] directly learns a spatial encoding of each point, and PointNet++ [39] extends it with a local feature extractor based on Farthest Point Sampling (FPS) and is trained with hierarchical feature learning architecture. There are other methods initially processing 2D images and then mapping the segmented 2D results onto the corresponding 3D coordinates of the point cloud. MVPNet [16] aggregates 2D multi-view image features into 3D point clouds, and then uses point-based networks to fuse features in 3D canonical space to predict 3D semantic labels. VMVF [25] selects various virtual views to render multiple 2D channels for training an effective 2D semantic segmentation model and then fuses features from these predictions onto the 3D mesh vertices to determine semantic labels. However, these methods depend on existing point cloud data as input, which limits their versatility for downstream tasks.

### 2.2 Nerf Segmentation

Semantic-NeRF [59] was the first to integrate semantics into NeRF by fusing noisy 2D segmentations into a consistent 3D model, improving segmentation accuracy and enabling novel view synthesis. Panoptic NeRF [10] and DM-NeRF [49] explore panoptic radiance fields for label transfer and scene editing, but both rely on manual ground truth annotations. Panoptic Neural Fields [24]

decomposes scenes into objects and backgrounds using instance-specific MLPs for objects and a shared MLP for the background, optimized jointly from color images and predicted segmentations. Several studies [19, 31] have explored the approach of lifting latent features from 2D foundation models [42] into 3D space to enable open-vocabulary text queries. Other approaches [44, 11, 52, 20] have demonstrated promising results in object-level segmentation tasks. Panoptic Lifting [46] and Contrastive Lift [1] generate 3D panoptic representations by lifting 2D machine-generated segmentation masks to 3D for multi-view consistency. While Panoptic Lifting [46] addresses inconsistencies in 2D instance identifiers through linear assignment, Contrastive Lift [1] uses contrastive clustering. Despite their success in multi-view consistent segmentation, NeRF-based methods are limited by slow rendering speeds and high memory usage during training due to their implicit nature.

### 2.3 Gaussian Segmentation

Segmenting the Gaussian field involves dividing it into distinct regions based on their properties, which is crucial for scene reconstruction and understanding. LangSplat [40], LEGaussians [45] and several works [5, 37, 29, 12, 7, 35, 41] incorporate language features from CLIP for open-world scene representation. SADG [27] specifically targets segmentation in dynamic scenes. For single-object segmentation, Gaussian-Cut [15] proposes a Gaussian distribution-based optimization framework, while GradiSeg [28] develops a novel gradient-driven segmentation approach. PLGS [51] adopts a methodology similar to Panoptic Lifting [46]. InstanceGaussian [26] and BCG [58] propose clustering-based methods for segmenting 3D Gaussian representations. SAGA [4] efficiently embeds 2D segmentation features into 3D Gaussian point features using contrastive learning. Feature 3DGS [60] enables 3D Gaussian splatting on semantic features via 2D foundation model distillation to extract arbitrary-dimension semantic features. Gaussian Grouping [56] and CoSSegGaussians [9] apply video segmentation methods to unify segmentation IDs from multiple views. However, video segmentation methods often fail when there are significant changes in viewing angles. OpenGaussian [53] achieves consistent 3D segmentation

through codebook discretization but cannot render precise 2D segmentations. Gaga [32] shares a similar motivation with our work. However, it does not explicitly consider the role of segmentation in Gaussian optimization, and the lack of segmentation-constrained optimization can result in meaningless floaters in the 3D segmentation field. Our approach overcomes these challenges by using point map fusion and plane-constrained Gaussian splatting to create a compact and consistent 3D Gaussian segmentation field.

## 3 Method

### 3.1 Preliminaries

3D Gaussian splatting [18] presents a powerful approach to scene representation, offering impressive reconstruction quality and faster rendering speeds compared to methods like Neural Radiance Field [33]. It achieves this by utilizing 3D Gaussians to explicitly represent scene geometry and appearance, resembling a point cloud. Each Gaussian is defined by its centroid  $\mathbf{x}$ , 3D covariance  $\Sigma$ , opacity  $\alpha$ , and color  $\mathbf{c}$ , represented as spherical harmonics (SH) coefficients of three degrees.

To effectively supervise these learnable attributes, Gaussian splatting projects them onto the 2D imaging plane for rendering RGB images from a given viewpoint. This projection is performed via  $\alpha$ -blending, a differentiable rasterization method optimized for GPU implementation. For each pixel, the color  $C$  is computed as follows:

$$C = \sum_{i \in \mathcal{N}} \mathbf{c}_i \alpha'_i \prod_{j=1}^{i-1} (1 - \alpha'_j) \quad (1)$$

Here,  $\mathbf{c}_i$  represents the color of the  $i$ -th Gaussian,  $\alpha'_i$  denotes its influence factor calculated by multiplying the projected 2D covariance with the per-point opacity  $\alpha_i$ .

To extend the Gaussian Rasterizer from color space  $C$  to segmentation space  $S$ , Identity Encoding  $\mathbf{s}_i$  [56] is introduced. The Identity Encoding is a learnable vector of length 16 that represents the segmentation feature for each Gaussian, enabling precise and differentiable instance segmentation within the Gaussian framework. For each pixel,

the segmentation  $S$  can be expressed as:

$$S = \sum_{i \in \mathcal{N}} \mathbf{s}_i \alpha'_i \prod_{j=1}^{i-1} (1 - \alpha'_j) \quad (2)$$

### 3.2 Pointmap-based Association for Consistent Segmentation

Mask association methods in video segmentation models often struggle with complex scenes involving multiple objects. To address these challenges, we propose a pointmap-based association approach to create a 3D segmentation field that aligns instance IDs across different viewpoints. By incorporating spatial information, our method achieves consistent and reliable multi-view segmentation.

The process begins with the creation of a unified 3D pointmap field. A pointmap  $P \in \mathbb{R}^{W \times H \times 3}$  represents a dense 2D field of 3D points, establishing a precise one-to-one correspondence between the pixels of an RGB image  $I$  with resolution  $W \times H$  and their respective 3D scene points. To achieve this, we employ DUSt3R [50] to construct a unified pointmap field  $\mathcal{P}$  that integrates individual pointmaps  $\{P_1, P_2, \dots, P_n\}$ . Subsequently, the images  $\{I_1, I_2, \dots, I_n\}$  are processed by a segmentation model [21] to generate a set of inconsistent masks  $\{M_1, M_2, \dots, M_n\}$ .

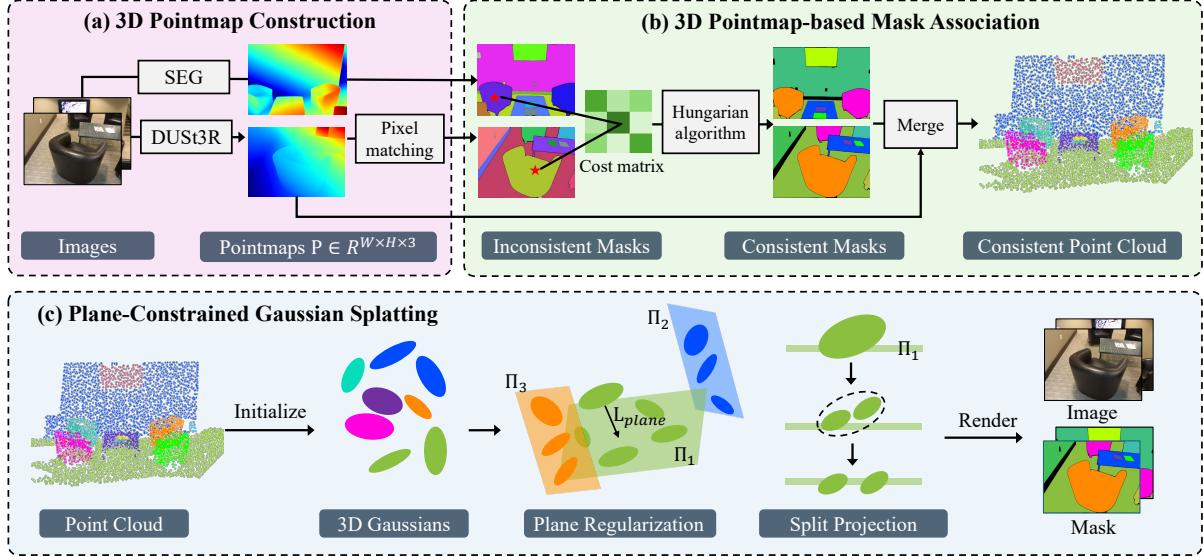
Given two adjacent images,  $I_t$  and  $I_{t+1}$ , along with their corresponding pointmaps,  $P_t$  and  $P_{t+1}$ , we establish pixel correspondence by measuring the distance between the pointmaps. For each pixel  $(i, j)$  in image  $I_t$ , we identify the closest pixel  $(k, l)$  in image  $I_{t+1}$  by minimizing the Euclidean distance between their respective pointmap values:

$$(k, l) = \arg \min_{(k', l')} \|P_t(i, j) - P_{t+1}(k', l')\|_2 \quad (3)$$

This correspondence is formally represented as a mapping function  $\phi : (i, j) \mapsto (k, l)$ , where

$$\phi(i, j) = (k, l) \quad (4)$$

For  $a^{th}$  object mask in  $M_t$  and  $b^{th}$  object mask in  $M_{t+1}$ , we redefine the overlap between two object masks  $M_t^a$  and  $M_{t+1}^b$  based on  $\phi$ :



**Fig. 2** The pipeline of our method. (a) We first construct a unified point cloud field and establish correspondences between pixels using the pointmaps. (b) Leveraging these relationships, we construct a cost matrix for instance masks across two frames. The Hungarian algorithm is then applied to optimize the cost matrix, ensuring consistent mask association. By merging all frames, we obtain a point cloud enriched with consistent segmentation information. (c) This point cloud serves as the initialization for 3D Gaussians. To achieve compact 3D segmentation, we employ a piecewise-plane constraint, restricting point displacement within local planes through plane regularization and split projection.

$$M_t^a \cap_{\phi} M_{t+1}^b = \{(i, j) \mid (i, j) \in M_t^a, \phi(i, j) \in M_{t+1}^b\} \quad (5)$$

This means that a pixel  $(i, j)$  contributes to the intersection only if it belongs to  $M_t^a$  and its corresponding pixel  $(k, l) = \phi(i, j)$  belongs to  $M_{t+1}^b$ . Denote  $|\cdot|$  as the number of points in a mask. Using this refined intersection definition, the matching cost between classes  $a$  and  $b$  is given by:

$$C(a, b) = 1 - \frac{|M_t^a \cap_{\phi} M_{t+1}^b|}{\min(|M_t^a|, |M_{t+1}^b|)} \quad (6)$$

The mask association problem can be effectively addressed using the Hungarian algorithm, which aims to minimize the total matching cost between objects in  $M_t$  and  $M_{t+1}$ . Specifically, let  $\mathbb{I}_{a,b} \in \{0, 1\}$  be a binary variable that indicates whether the object  $M_t^a$  in the current frame is matched with the object  $M_{t+1}^b$  in the next frame. The objective is to determine the optimal segmentation correspondence function  $\psi : \mathcal{A} \mapsto \mathcal{B}$ ,

which assigns object in  $M_t$  to its most appropriate counterpart in  $M_{t+1}$  by minimizing the total association cost. The objective function for this optimization problem is defined as:

$$\psi = \arg \min_{\psi} \sum_{a \in \mathcal{A}} \sum_{b \in \mathcal{B}} C(a, b) \cdot \mathbb{I}_{a,b} \quad (7)$$

The constraints are relaxed to allow partial matching:

$$\begin{cases} \sum_{b \in \mathcal{B}} \mathbb{I}_{a,b} \leq 1, & \forall a \in \mathcal{A} \\ \sum_{a \in \mathcal{A}} \mathbb{I}_{a,b} \leq 1, & \forall b \in \mathcal{B} \\ \mathbb{I}_{a,b} \in \{0, 1\}, & \forall a \in \mathcal{A}, \forall b \in \mathcal{B} \end{cases} \quad (8)$$

Through this matching process, we obtain a set of consistently labeled 2D masks,  $\mathbf{m}$ , for all multi-view images, along with a 3D segmented point cloud field,  $\mathcal{P}^s$ . These unified 2D masks serve as pseudo-labels for the subsequent training of the Gaussian field, ensuring cross-view consistency and alignment. Meanwhile, the 3D segmented point cloud field  $\mathcal{P}^s$ , constructed by progressively integrating associated points, is utilized to initialize the Gaussian splatting process.

### 3.3 Piecewise-Plane Constrained Gaussian Splatting

To ensure compact 3D segmentation during training, we introduce a piecewise-plane constraint. For each point position  $\mathbf{x}_i$  in the Gaussian field  $\mathcal{G}$ , its nearest neighbors  $\mathcal{N}(\mathbf{x}_i)$  of the same class are used to fit a local plane  $\Pi_i$ . The constraint is then formulated to minimize the distance between each point  $\mathbf{x}_i$  and its corresponding local plane  $\Pi_i$ . By enforcing this constraint, points are encouraged to remain close to their class-specific neighborhood planes during optimization and densification, thus maintaining the structural integrity and compactness of the 3D segmentation. This approach provides two key benefits: first, the plane constraint during optimization reduces the occurrence of floating points, ensuring that all points adhere to the object's surface, thereby preserving the compactness of the 3D segmentation field. Second, during densification, replicated points are restricted to the local neighborhood of similar points, minimizing ambiguity and misclassification at object boundaries. To simplify computation, the piecewise-planes are updated every 1000 iterations.

#### Plane Regularization

To implement the piecewise-plane constraint, we minimize the distance of points to their corresponding planes. For a given point position  $\mathbf{x}_i$ , let  $(a_i, b_i, c_i)$  represent the coefficients of the normal vector  $\mathbf{n}_i$  of its neighborhood plane. The equation of the neighborhood plane  $\Pi_i$  can be expressed as:

$$\Pi_i : a_i x + b_i y + c_i z + D_i = 0 \quad (9)$$

An arbitrary point  $\mathbf{x}_p$  lying on the plane  $\Pi_i$  is used to compute the perpendicular distance from  $\mathbf{x}_i$  to the plane. The distance is calculated as:

$$d_i = |\mathbf{n}_i^T \cdot (\mathbf{x}_i - \mathbf{x}_p)| \quad (10)$$

To enforce this constraint across the entire 3D Gaussian field, we define a plane regularization loss as the average distance of all points to their respective planes. Given  $r$  Gaussian points, the loss is expressed as:

$$\mathcal{L}_{\text{plane}} = \frac{1}{r} \sum_{i=1}^r d_i. \quad (11)$$

As illustrated in Figure 2, incorporating this plane regularization loss ensures that points are primarily adjusted within their respective categories, improving the coherence and precision of the segmentation and reconstruction process.

#### Split projection

The adaptive control process identifies points with excessively large gradients in the Gaussian position  $\mathbf{x}$  for densification. At these points, smaller Gaussians are cloned, while larger Gaussians are split. During the cloning or splitting process of Gaussians, we replicate their original class assignments to ensure that points copied from one class are not optimized as belonging to another. This prevents boundary confusion among Gaussians. Ultimately, by following the same-class cloning rule, we obtain  $\mathcal{P}^{s'}$  from the initial segmentation field  $\mathcal{P}^s$ . We denote the cloned Gaussians position as  $\mathbf{x}^c$  and the split Gaussians position as  $\mathbf{x}^s$ . The cloned Gaussians  $\mathbf{x}^c$  replicate the original position and shifts in the direction of the positional gradient, with plane optimization constraining its new location. For split Gaussians  $\mathbf{x}^s$ , the positions of the newly generated points are determined by sampling from the original 3D Gaussian distribution. The original 3D Gaussian is treated as a probability density function, guiding the placement of these new points. However, this sampling-based initialization may cause the new points to deviate from the intended piecewise-plane structure. Consequently, different classes tend to mix at the boundaries after splitting. To prevent split Gaussians from being optimized as other categories, we project the split points onto the piecewise-plane. If  $\mathbf{x}_i^s$  is the point to be projected, the vertical distance from the point to the plane is  $d_i$ , and the split projection operation, which maps the point  $\mathbf{x}_i^s$  onto the plane, can be defined as:

$$\mathbf{x}_i^{s'} = \mathbf{x}_i^s - d_i \mathbf{n}_i \quad (12)$$

#### Training objective

Similar to [56], we use a linear layer  $f$  followed by a softmax function to map the rendered 2D features  $S$  in Equation (2) to a  $K$  classification space. For this 2D classification, we employ a standard cross-entropy loss  $\mathcal{L}_{2d}$ , which ensures accurate

---

**Algorithm 1** Pseudocode for CCGS

---

**Require:** Input images  $\mathcal{I} = \{I_1, I_2, \dots, I_n\}$ , Consistent masks  $\mathbf{m} = \emptyset$ , Segmented point cloud  $\mathcal{P}^s = \emptyset$

# Get inconsistent segmentation  
 $\{M_1, M_2, \dots, M_n\} \leftarrow \text{SEG}(\mathcal{I})$

# Get Pointmap  
 $\{P_1, P_2, \dots, P_n\} \leftarrow \text{DUST3R}(\mathcal{I})$

$m_1 \leftarrow M_1$

for  $t \in \{1, 2, \dots, n-1\}$  do

# Establish pixel correspondence  
 for  $(i, j) \in I_t$  do  
 $\phi(i, j) \leftarrow \arg \min \|P_t(i, j) - P_{t+1}(k', l')\|_2$   
 end for

# Compute mask overlap and cost  
 $C(a, b) \leftarrow 1 - \frac{|M_t^a \cap \phi M_{t+1}^b|}{\min(|M_t^a|, |M_{t+1}^b|)}$

# Solve mask association  
 $\psi \leftarrow \text{Hungarian}(C)$   
 $m_{t+1} \leftarrow M_{t+1}^{\psi(m_t)}$

# Get segmented point cloud  
 $P_{t+1}^s = \text{cat}(m_{t+1}, P_{t+1})$   
 $\mathbf{m} \leftarrow \mathbf{m} \cup m_{t+1}, \mathcal{P}^s \leftarrow \mathcal{P}^s \cup P_{t+1}^s$

end for

# Piecewise-plane constrained gaussian splatting

while not converged do

if IsUpdatePlaneIteration then  
 $\Pi \leftarrow \text{CalculatePiecewise-Plane}(\mathbf{x}, s)$   
 end if

$\hat{I}, \hat{\mathbf{m}} \leftarrow \text{Rasterize}(\mathbf{x}, s, \Sigma, \mathbf{c}, \alpha)$   
 $\mathcal{L}_{\text{img}} \leftarrow \mathcal{L}(I, \hat{I})$   
 $\mathcal{L}_{\text{2d}} \leftarrow \mathcal{L}(\mathbf{m}, \hat{\mathbf{m}}), \mathcal{L}_{\text{3d}} \leftarrow \mathcal{L}(\mathcal{P}^s, s)$

# Plane Regularization  
 $\mathcal{L}_{\text{plane}} \leftarrow \mathcal{L}(\Pi, \mathbf{x})$   
 $\mathcal{L}_{\text{render}} \leftarrow \mathcal{L}_{\text{img}} + \mathcal{L}_{\text{2d}} + \mathcal{L}_{\text{3d}} + \mathcal{L}_{\text{plane}}$   
 $\mathbf{x}, s, \Sigma, \mathbf{c}, \alpha \leftarrow \text{Adam}(\nabla \mathcal{L}_{\text{render}})$

if IsRefinementIteration then

# Densification  
 $\mathbf{x}^s, \mathbf{x}^c \leftarrow \text{D}(\mathbf{x})$   
# Split Projection  
 $\mathbf{x}^s \leftarrow \mathbf{x}^s - d\mathbf{n}$

end if

end while

---

pixel-level mask predictions.

$$\mathcal{L}_{\text{2d}} = - \sum_{k \in \mathcal{K}} \mathbf{m}[k] \log(\text{softmax}(f(S))[k]) \quad (13)$$

To maintain the consistency of 3D segmentation, we also imposed cross-entropy constraints  $\mathcal{L}_{\text{3d}}$  on the 3D segmentation features  $s$  of each point, using the 3D segmentation merged from the pointmap as a pseudo label.

$$\mathcal{L}_{\text{3d}} = - \sum_{k \in \mathcal{K}} \mathcal{P}^s[k] \log(\text{softmax}(f(s))[k]) \quad (14)$$

Combined with the conventional 3D Gaussian Loss  $\mathcal{L}_{\text{img}} = (1-\lambda)\mathcal{L}_1 + \lambda \mathcal{L}_{\text{D-SSIM}}$  on image rendering [18], the total loss  $\mathcal{L}_{\text{render}}$  for fully end-to-end training is:

$$\mathcal{L}_{\text{render}} = \mathcal{L}_{\text{img}} + \lambda_{\text{plane}} \mathcal{L}_{\text{plane}} + \lambda_{\text{2d}} \mathcal{L}_{\text{2d}} + \lambda_{\text{3d}} \mathcal{L}_{\text{3d}} \quad (15)$$

The final pseudocode is presented in Algorithm 1.

## 4 Experiment

### 4.1 Experimental Setup

#### Datasets

We present experimental results on two datasets: Replica [47] and ScanNet [8]. The Replica Dataset consists of high-quality reconstructions of various indoor scenes, with each scene containing RGB images paired with corresponding 2D segmentation masks. ScanNet is a large-scale real-world dataset, where each scene comprises images accompanied by annotated 2D segmentation masks. For both datasets, we select 7 scenes for training and evaluation, following a similar approach to [9]. Each scene contains approximately 200 training images and 50 testing images, which are uniformly sampled from the dataset. To generate training labels, we utilize SAM [21], which produces high-quality object masks. The annotated segmentation masks provided by the dataset are only available during evaluation as ground-truth labels and are not used during training.

#### Data availability statement

The Replica dataset (<https://doi.org/10.48550/arXiv.1906.05797>) and can be accessed at <https://github.com/facebookresearch/Replica-Dataset>. We use pre-rendered Replica dataset provided by Semantic-NeRF (<https://doi.org/10.48550/arXiv.2103.15875>). The ScanNet dataset

(<https://doi.org/10.48550/arXiv.1702.04405>) and can be accessed at <http://www.scan-net.org>.

### Evaluation metrics

For 2D segmentation, we use mean intersection over union (mIoU) to evaluate the quality of predicted masks. In a single-view setting,  $mIoUs$  is computed by averaging IoU values across all predicted and ground-truth masks, with optimal assignments determined via linear sum assignment. For multi-view segmentation, we construct a global IoU matrix by aggregating IoU values of masks with the same ID across different viewpoints. The final  $mIoUm$  is computed from these aggregated values, providing a unified evaluation across multiple views. Additionally, We calculate PSNR and SSIM to evaluate the quality of rendered images. For 3D Gaussian segmentation, we abstract Gaussians into a segmentation field composed of spatial coordinates  $x$  and segmentation features  $s$ . To evaluate the segmentation quality, we first align the ground truth point cloud segmentation and reconstructed Gaussian fields to the same scale and orientation. The ground truth labels are then mapped onto the reconstructed field using a nearest neighbor approach. If the nearest distance exceeds a threshold  $\gamma$ , the point is assigned as ‘no category’, introducing a penalty for floaters in free space. We then compute the  $mIoU_{3D}$  on the reconstructed segmentation field as the primary evaluation metric. Additionally, to further assess the geometric fidelity of the reconstruction, we employ Chamfer Distance to quantify the structural similarity between the ground truth and the reconstructed Gaussian field.

### Implementation details

We implement our method based on Gaussian Grouping [56]. For a comprehensive and fair comparison, our method, along with Gaussian Grouping and OpenGaussian, operates on the same initial point cloud for 3D Gaussian segmentation and various downstream tasks. The threshold parameters,  $\gamma$  is set to 0.5. During training, we set  $\lambda_{\text{plane}} = 10$ ,  $\lambda_{2d} = 1$ , and  $\lambda_{3d} = 1$ . The piecewise-plane is estimated using the 10 nearest neighbors. We utilize the Adam optimizer to update both Gaussian parameters, with the learning rates for Gaussians identical to those used in the original Gaussian Splatting. All datasets are trained and

evaluated for 30K iterations on a single NVIDIA 4090 GPU.

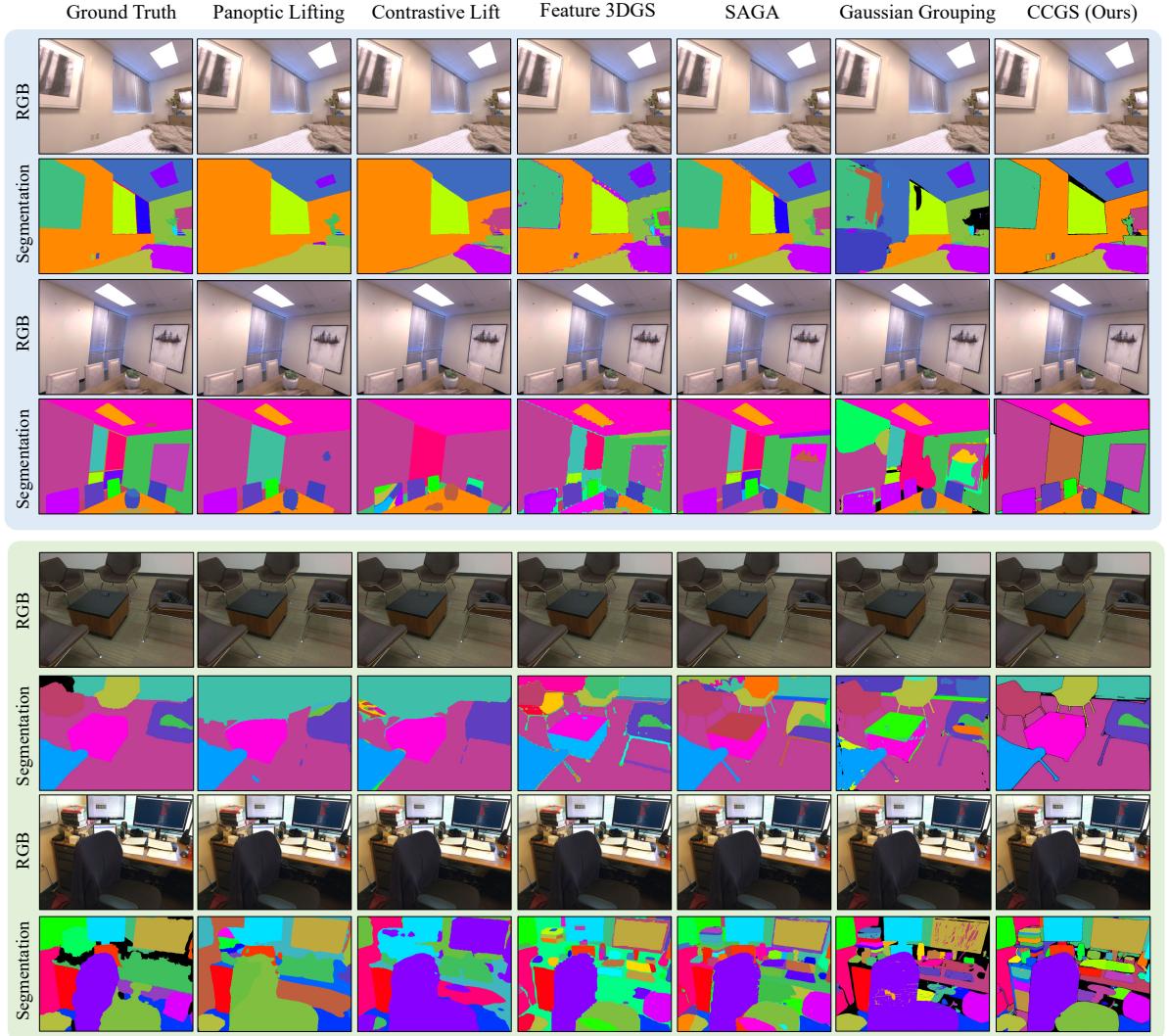
## 4.2 Experimental Results

### 2D segmentation

To validate CCGS on 2D panoptic segmentation task, we compare the results of Panoptic Lifting [46], Contrastive Lift [1], SAGA [13], Feature 3DGS [60] and Gaussian Grouping [56] with CCGS on the Replica and ScanNet dataset. Since SAGA and Feature 3DGS are designed only for single-view segmentation tasks and cannot provide consistent segmentation IDs across multiple views, we match the segmentation IDs from their single-view results to the IDs generated by video segmentation methods [6]. This serves as an extension of the video mask association approach.

As shown in Table 1, for single view  $mIoUs$ , our CCGS method achieves better performance compared to Gaussian Grouping with improvements of 2.38% on the ScanNet dataset and 1.29% on the Replica dataset. By comparing the single-view  $mIoUs$  and multi-view  $mIoUm$ , our method experiences a 3% drop in IoU, which is attributed to mismatches caused by differences in segmentation scales across views. Panoptic Lifting and Contrastive Lift also show a relatively small IoU drop of around 5-8%, indicating that their methods maintain multi-view consistency. However, video mask association-based methods, such as Gaussian Grouping, Feature 3DGS, and SAGA, suffer a much larger IoU drop of about 20%, which strongly suggests that video segmentation methods lack multi-view consistency in datasets with significant view differences. When compared to Gaussian Grouping on the Replica dataset, CCGS shows enhancements of 1.15 in PSNR, and 0.031 in SSIM. On the ScanNet dataset, CCGS similarly surpasses Gaussian Grouping by 1.05 in PSNR, and 0.052 in SSIM. The increase in PSNR and SSIM indicates that while achieving excellent 2D segmentation, the introduction of piecewise-plane constraint also helps to better reconstruct the entire scene.

The qualitative results on the Replica and ScanNet datasets are presented in Figure 3. Panoptic Lifting exhibits issues with incomplete segmentation and blurred boundaries. Contrastive Lift struggles to maintain training stability in complex scenarios, often leading to

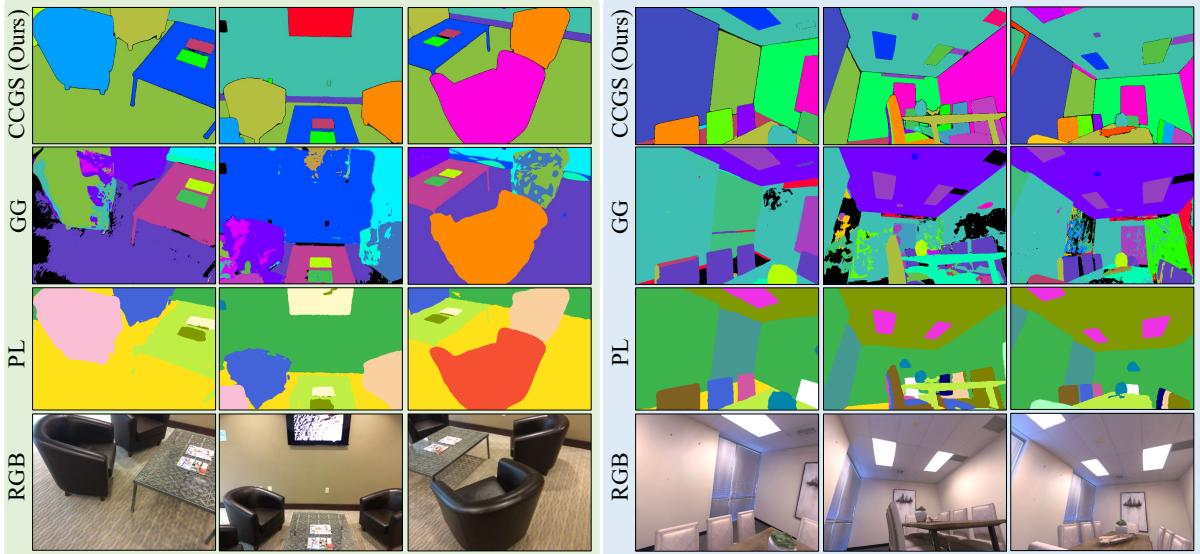


**Fig. 3** 2D segmentation results on Replica and ScanNet datasets. Each column from left to right in the figure represents Ground truth segmentation, Panoptic Lifting, Contrastive Lift, Feature 3DGS, SAGA, Gaussian Grouping and Ours (CCGS). The top four lines represent different scenes in Replica. The following four lines are from different scenes in ScanNet.

over-segmentation in the later stages of training. FeatureGS, on the other hand, suffers from feature confusion and uneven edges at object boundaries. This issue arises because FeatureGS only aligns SAM features with Gaussian point features at the 2D level without incorporating actual masks during training. SAGA demonstrates strong performance in 2D segmentation. However, it is inherently unsuitable for obtaining multi-view consistent masks. Due to the lack of spatial information in video segmentation

method, Gaussian Grouping exhibits inconsistencies across multiple perspectives, leading to over-segmentation and numerous artifacts in the segmentation results during rendering. In contrast, CCGS, with pointmap fusion ensuring consistent segmentation, outperforms other methods in both segmentation completeness and accuracy.

Figure 4 presents the visualization results of multi-view consistency. While Panoptic Lifting achieves multi-view consistent segmentation, the quality of the segmentation boundaries remains



**Fig. 4** The comparison on multi-view consistency between CCGS and GG (Gaussian Grouping) and PL (Panoptic Lifting). From top to bottom, the images display CCGS, GG, PL and RGB inputs, respectively.

**Table 1** The 2D panoptic segmentation results on the Replica and ScanNet datasets demonstrate that CCGS outperforms other methods on both datasets.

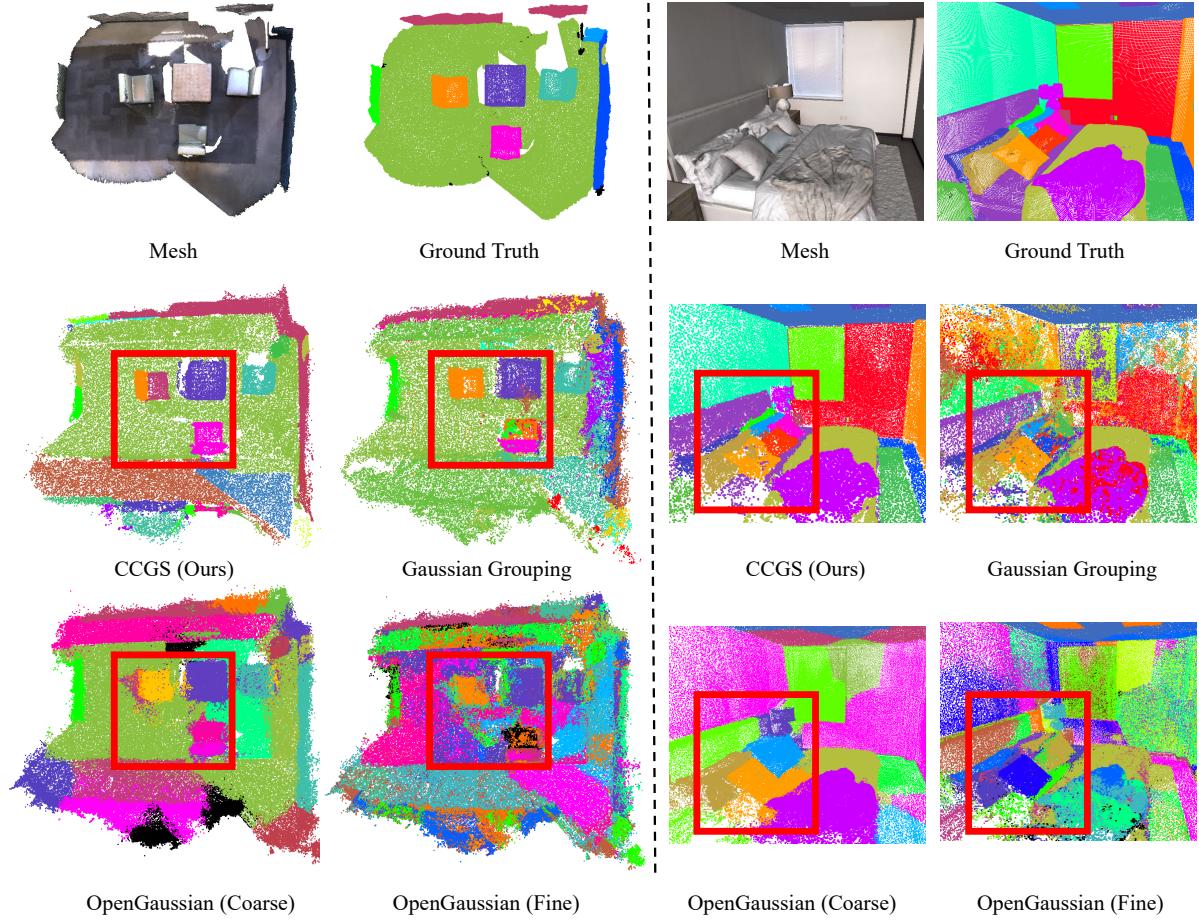
Model	Replica				ScanNet			
	$mIoU_s$	$mIoU_m$	PSNR	SSIM	$mIoU_s$	$mIoU_m$	PSNR	SSIM
Panoptic Lifting [46]	62.56	55.75	30.23	0.892	57.24	52.33	26.23	0.812
Contrastive Lift [1]	60.88	52.93	30.26	0.901	55.30	49.61	26.18	0.813
Feature 3DGS [60]	63.84	43.51	32.31	0.926	58.08	40.37	27.25	0.882
SAGA [13]	64.86	45.32	32.50	0.939	62.62	42.54	27.44	0.890
Gaussian Grouping [56]	64.25	47.16	32.41	0.935	61.54	44.37	27.27	0.889
CCGS (ours)	<b>65.54</b>	<b>62.31</b>	<b>33.56</b>	<b>0.966</b>	<b>63.92</b>	<b>60.27</b>	<b>28.32</b>	<b>0.941</b>

relatively low. Gaussian Grouping tends to produce more artifacts when applied to datasets where objects are not centrally positioned. For instance, the same chair in Gaussian Grouping appears in different colors across various views, and the wall is over-segmented due to inconsistent IDs. In contrast, objects segmented by CCGS not only maintain consistent segmentation across different views but also preserve sharper boundaries and more coherent object structures.

### 3D Gaussian segmentation

To assess CCGS for the 3D Gaussian segmentation task, we compare its performance with that of Gaussian Grouping [56] and OpenGaussian [53] on the Replica and ScanNet datasets.

OpenGaussian focuses on point-level 3D understanding and proposes a two-stage codebook to discretize features from a coarse to a fine level. As detailed in Table 2, our CCGS method surpasses Gaussian Grouping on the Replica dataset with improvements of 11.34% in  $mIoU_{3D}$ , and 0.038 in Chamfer Distance. Similarly, on the ScanNet dataset, CCGS exceeds Gaussian Grouping by 11.17% in  $mIoU_{3D}$ , and 0.031 in Chamfer Distance. The difference between CCGS and GG in  $mIoU_{3D}$  is significantly greater than in  $mIoU_s$  in Table 1. This discrepancy arises because  $mIoU_s$  does not account for multi-view consistency and only evaluates IoU for individual images and ground truth. At the coarse level, OpenGaussian and Gaussian Grouping achieve comparable



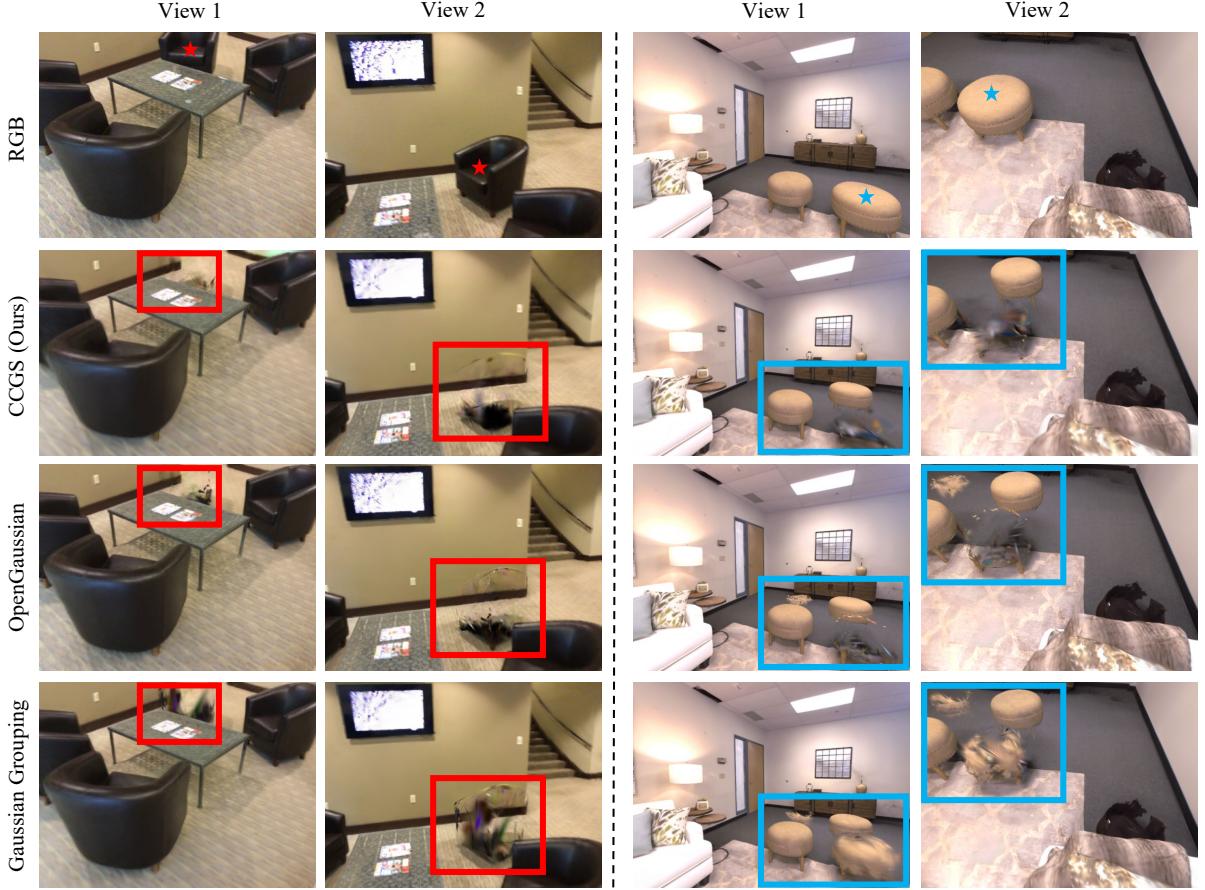
**Fig. 5** 3D Gaussian segmentation results on ScanNet and Replica datasets. Each scene consists of a ground truth mesh, ground truth point cloud segmentation, our method (CCGS), Gaussian Grouping, as well as coarse-level and fine-level OpenGaussian results.

**Table 2 3D Gaussian segmentation** results on Replica and ScanNet datasets.

Model	Replica		ScanNet	
	$mIoU_{3D} \uparrow$	Chamfer Distance $\downarrow$	$mIoU_{3D} \uparrow$	Chamfer Distance $\downarrow$
OpenGaussian(Fine) [53]	20.56	0.237	23.69	0.496
OpenGaussian(Coarse) [53]	53.34	0.237	50.56	0.496
Gaussian Grouping [56]	54.12	0.230	52.04	0.482
CCGS (ours)	<b>65.46</b>	<b>0.192</b>	<b>63.21</b>	<b>0.451</b>

$mIoU_{3D}$ . However, at the fine level,  $mIoU_{3D}$  of OpenGaussian is significantly lower. Meanwhile, the reduction in Chamfer Distance indicates that plane-constrained Gaussian splatting optimizes Gaussians with enhanced structural properties.

As shown in the red box of the left scene in Figure 5, both Gaussian Grouping and coarse-level OpenGaussian exhibit floaters. This issue is particularly evident in OpenGaussian, where yellow points originally belonging to the chair are propagated onto the purple coffee table, leading to



**Fig. 6** Results of the downstream deletion and movement tasks. The first row represents the RGB image, where  $\star$  marks the same objects to be deleted from different viewpoints, and  $\star$  marks the same objects to be moved from different viewpoints.

classification boundary confusion and disrupting object integrity. In contrast, our CCGS method, leveraging plane regularization and split projection, effectively constrains the optimization of same category points within piecewise-planes. As a result, our approach achieves clear boundaries and well-structured reconstruction, significantly reducing floaters and improving spatial coherence. In the right-side scene, Gaussian Grouping exhibits noticeable color bleeding, where accumulated 2D inconsistencies lead to significant 3D inconsistencies. Coarse-level OpenGaussian fails to separate similar pillows, resulting in under-segmentation that merges distinct objects into a single cluster, while fine-level OpenGaussian suffers from over-segmentation. In contrast, our CCGS method achieves consistent segmentation with clear object boundaries.

#### Downstream tasks

As shown in Figure 6, in the deletion task, Gaussian grouping and OpenGaussian leaves many undeleted points due to inconsistent IDs. In contrast, our method effectively deletes Gaussian points, leaving minimal residue. In the movement task, both Gaussian Grouping and OpenGaussian cause parts of adjacent chairs to be moved as well. This occurs because the chairs are positioned closely together, and during optimization and cloning, category points from one chair are propagated to nearby similar chairs, leading to confusion between categories. In contrast, our method ensures that the movement of one chair does not affect other objects in the scene. Meanwhile, the ground after the movement is free from artifacts caused by residual points, unlike in Gaussian Grouping and OpenGaussian.

**Table 3** The comparison of different components PF (Pointmap Fusion), PR (Plane Regularization), SP (Split Projection) of CCGS on the ScanNet dataset.

PF	PR	SP	$mIoU_s \uparrow$	$mIoU_m \uparrow$	$mIoU_{3D} \uparrow$	Chamfer Distance $\downarrow$
-	-	-	61.54	44.37	52.04	0.482
✓	-	-	63.38	60.18	57.12	0.480
✓	✓	-	63.83	60.21	59.56	0.469
✓	✓	✓	<b>63.92</b>	<b>60.27</b>	<b>63.21</b>	<b>0.451</b>

### 4.3 Ablation Study

#### Pointmap fusion

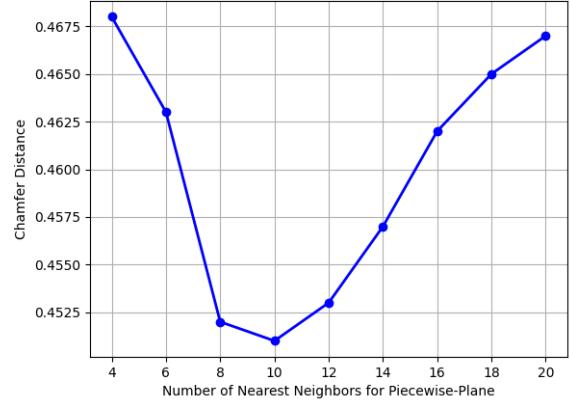
As shown in Table 3, incorporating pointmap fusion improves single-view  $mIoU_s$  by 1.84%, multi-view  $mIoU_m$  by 15.81%,  $mIoU_{3D}$  by 5.08%, and Chamfer Distance by 0.002. This demonstrates that pointmap fusion significantly enhances multi-view consistency, as reflected in the substantial improvements in  $mIoU_m$  and  $mIoU_{3D}$ . These results highlight the superiority of pointmap fusion over traditional video segmentation methods in maintaining multi-view consistency. Meanwhile, the Chamfer Distance remains relatively unchanged, as pointmap fusion primarily aligns 2D segmentation pseudo-labels without modifying the underlying 3D Gaussian structure during training.

#### Plane regularization

Plane regularization improves  $mIoU_s$ ,  $mIoU_m$ ,  $mIoU_{3D}$ , and Chamfer Distance by 0.45%, 0.03%, 2.44%, and 0.011, respectively. Since we treat points that are far from the ground truth as unlabeled when calculating  $mIoU_{3D}$ , the compactness of the reconstruction directly contributes to the improvement in  $mIoU_{3D}$ . This demonstrates that plane regularization helps better preserve the movement of Gaussians within the plane of similar points, thereby enhancing the robustness of the reconstruction.

#### Split projection

Split projection further enhances the performance of CCGS. As shown in Table 3, incorporating split projection improves  $mIoU_s$ ,  $mIoU_m$ ,  $mIoU_{3D}$ , and Chamfer Distance by 0.09%, 0.06%, 3.65%, and 0.018, respectively. Compared to plane regularization, split projection is more effective in improving  $mIoU_{3D}$ . This suggests that the initialization of Gaussian positions during the split



**Fig. 7** Chamfer Distance vs. Number of Nearest Neighbors for Piecewise-Plane

process introduces greater uncertainty in the segmentation of the 3D field. By projecting the initialized points onto the piecewise-plane, split projection enhances object compactness and reduces confusion at the boundaries.

#### Nearest neighbors for piecewise-plane

We conducted experiments on ScanNet, using 4 to 20 nearest neighbors to construct the piecewise-plane, selecting the optimal number based on Chamfer Distance. As shown in Figure 7, using only 4 points results in a higher Chamfer Distance. This is because a plane constructed with just 4 points cannot accurately represent the sub-plane in areas with dense Gaussian points. As the number of neighbors increases, Chamfer Distance decreases, reaching its minimum at 10 points. However, beyond this point, the distance starts to rise again. This happens because using too many neighbors causes regions that should be curved to become flattened, leading to deformation in some points. These findings highlight the importance of balancing local geometric fidelity and global structural integrity when constructing piecewise-planes

for Gaussian optimization, ensuring accurate surface representation while preserving the overall scene structure.

## 5 Limitations

Despite the promising results of CCGS, there are some limitations to consider. The quality of 2D segmentation may degrade in highly occluded or fast-moving viewpoints. While CCGS helps alleviate some of these issues, they can still negatively impact the overall segmentation performance. Future work could focus on optimizing 2D pseudo-labels during training, perhaps by incorporating more sophisticated temporal consistency constraints. Additionally, the current CCGS segmentation method is limited to static 3D scenes. Expanding the approach to dynamic 3D Gaussian segmentation, where objects can move and interact over time, will be an important direction for further research.

## 6 Conclusion

We introduce CCGS, a consistent and compact 3D Gaussian segmentation field that significantly improves segmentation quality. By constructing a unified 3D field through pointmap fusion, CCGS effectively addresses inconsistencies caused by occlusions and viewpoint changes, ensuring reliable segmentation even in challenging scenarios. Additionally, we employ plane-constrained Gaussian splatting to ensure that points remain within their respective piecewise-planes, preventing the creation of ambiguous category points and improving segmentation clarity. Overall, our method substantially enhances the quality and consistency of segmentation results in both 2D and 3D domains, offering promising applications in a wide range of scene understanding, manipulation, and editing tasks.

## References

- [1] Y. Bhalgat, I. Laina, J. F. Henriques, A. Zisserman, and A. Vedaldi. Contrastive lift: 3d object instance segmentation by slow-fast contrastive fusion. *arXiv preprint arXiv:2306.04633*, 2023.
- [2] H. Bian, L. Kong, H. Xie, L. Pan, Y. Qiao, and Z. Liu. Dynamicccity: Large-scale lidar generation from dynamic scenes. *arXiv preprint arXiv:2410.18084*, 2024.
- [3] M. Castillo, M. Dahaghin, M. Toso, and A. Del Bue. Contrastive gaussian clustering for weakly supervised 3d scene segmentation. In *International Conference on Pattern Recognition*, pages 114–130. Springer, 2024.
- [4] 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.
- [5] R. Chen, X. Sun, Z. Wang, Y. Liu, J. Wang, L. Kong, J. Deng, M. Gong, L. Pan, W. Wang, et al. Ovgaussian: Generalizable 3d gaussian segmentation with open vocabularies. *arXiv preprint arXiv:2501.00326*, 2024.
- [6] H. K. Cheng, S. W. Oh, B. Price, A. Schwing, and J.-Y. Lee. Tracking anything with decoupled video segmentation. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 1316–1326, 2023.
- [7] J. Cheng, J.-N. Zaech, L. Van Gool, and D. P. Paudel. Occam’s lgs: A simple approach for language gaussian splatting. *arXiv preprint arXiv:2412.01807*, 2024.
- [8] A. Dai, A. X. Chang, M. Savva, M. Halber, T. Funkhouser, and M. Nießner. ScanNet: Richly-annotated 3d reconstructions of indoor scenes. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 5828–5839, 2017.
- [9] B. Dou, T. Zhang, Y. Ma, Z. Wang, and Z. Yuan. Cosseggaussians: Compact and swift scene segmenting 3d gaussians. *arXiv preprint arXiv:2401.05925*, 2024.
- [10] X. Fu, S. Zhang, T. Chen, Y. Lu, L. Zhu, X. Zhou, A. Geiger, and Y. Liao. Panoptic nerf: 3d-to-2d label transfer for panoptic urban scene segmentation. In *2022 International Conference on 3D Vision (3DV)*, pages 1–11. IEEE, 2022.
- [11] R. Goel, D. Sirikonda, S. Saini, and P. Narayanan. Interactive segmentation of radiance fields. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 4201–4211, 2023.
- [12] J. Hu, Z. Chen, Z. Li, Y. Xu, and J. Zhang. Sparselgs: Sparse view language embedded gaussian splatting. *arXiv preprint arXiv:2401.05925*, 2024.

- arXiv:2412.02245*, 2024.
- [13] X. Hu, Y. Wang, L. Fan, J. Fan, J. Peng, Z. Lei, Q. Li, and Z. Zhang. Semantic anything in 3d gaussians. *arXiv preprint arXiv:2401.17857*, 2024.
- [14] R. Huang, S. Peng, A. Takmaz, F. Tombari, M. Pollefeys, S. Song, G. Huang, and F. Engelmann. Segment3d: Learning fine-grained class-agnostic 3d segmentation without manual labels. In *European Conference on Computer Vision*, pages 278–295. Springer, 2024.
- [15] U. Jain, A. Mirzaei, and I. Gilitschenski. Gaussiancut: Interactive segmentation via graph cut for 3d gaussian splatting. In *The Thirty-eighth Annual Conference on Neural Information Processing Systems*, 2024.
- [16] M. Jaritz, J. Gu, and H. Su. Multi-view pointnet for 3d scene understanding. In *Proceedings of the IEEE/CVF international conference on computer vision workshops*, pages 0–0, 2019.
- [17] A. Kamran-Pishhesari, A. Moniri-Morad, and J. Sattarvand. Applications of 3d reconstruction in virtual reality-based teleoperation: A review in the mining industry. *Technologies*, 12(3):40, 2024.
- [18] B. Kerbl, G. Kopanas, T. Leimkühler, and G. Drettakis. 3d gaussian splatting for real-time radiance field rendering. *ACM Transactions on Graphics*, 42(4):1–14, 2023.
- [19] 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*, pages 19729–19739, 2023.
- [20] C. M. Kim, M. Wu, J. Kerr, K. Goldberg, M. Tancik, and A. Kanazawa. Garfield: Group anything with radiance fields. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 21530–21539, 2024.
- [21] A. Kirillov, E. Mintun, N. Ravi, H. Mao, C. Rolland, L. Gustafson, T. Xiao, S. Whitehead, A. C. Berg, W.-Y. Lo, et al. Segment anything. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 4015–4026, 2023.
- [22] L. Kong, Y. Liu, X. Li, R. Chen, W. Zhang, J. Ren, L. Pan, K. Chen, and Z. Liu. Robo3d: Towards robust and reliable 3d perception against corruptions. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 19994–20006, 2023.
- [23] L. Kong, X. Xu, J. Ren, W. Zhang, L. Pan, K. Chen, W. T. Ooi, and Z. Liu. Multi-modal data-efficient 3d scene understanding for autonomous driving. *arXiv preprint arXiv:2405.05258*, 2024.
- [24] A. Kundu, K. Genova, X. Yin, A. Fathi, C. Pantofaru, L. J. Guibas, A. Tagliasacchi, F. Dellaert, and T. Funkhouser. Panoptic neural fields: A semantic object-aware neural scene representation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 12871–12881, 2022.
- [25] A. Kundu, X. Yin, A. Fathi, D. Ross, B. Brewington, T. Funkhouser, and C. Pantofaru. Virtual multi-view fusion for 3d semantic segmentation. In *Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part XIV 16*, pages 518–535. Springer, 2020.
- [26] H. Li, Y. Wu, J. Meng, Q. Gao, Z. Zhang, R. Wang, and J. Zhang. Instancegaussian: Appearance-semantic joint gaussian representation for 3d instance-level perception. *arXiv preprint arXiv:2411.19235*, 2024.
- [27] Y.-J. Li, M. Gladkova, Y. Xia, and D. Creemers. Sadg: Segment any dynamic gaussian without object trackers. *arXiv preprint arXiv:2411.19290*, 2024.
- [28] Z. Li, W. Han, Y. Cai, H. Jiang, B. Bi, S. Gao, H. Zhao, and Z. Wang. Gradiseg: Gradient-guided gaussian segmentation with enhanced 3d boundary precision. *arXiv preprint arXiv:2412.00392*, 2024.
- [29] S. Liang, S. Wang, K. Li, M. Niemeyer, S. Gasperini, N. Navab, and F. Tombari. Supergseg: Open-vocabulary 3d segmentation with structured super-gaussians. *arXiv preprint arXiv:2412.10231*, 2024.
- [30] G. Liu, O. van Kaick, H. Huang, and R. Hu. Active self-training for weakly supervised 3d scene semantic segmentation. *Computational Visual Media*, 10(3):425–438, 2024.
- [31] K. Liu, F. Zhan, J. Zhang, M. Xu, Y. Yu, A. El Saddik, C. Theobalt, E. Xing, and S. Lu. Weakly supervised 3d open-vocabulary

- segmentation. *Advances in Neural Information Processing Systems*, 36:53433–53456, 2023.
- [32] W. Lyu, X. Li, A. Kundu, Y.-H. Tsai, and M.-H. Yang. Gaga: Group any gaussians via 3d-aware memory bank. *arXiv preprint arXiv:2404.07977*, 2024.
- [33] 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. *Communications of the ACM*, 65(1):99–106, 2021.
- [34] C. Park, Y. Jeong, M. Cho, and J. Park. Fast point transformer. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 16949–16958, 2022.
- [35] Q. Peng, B. Planche, Z. Gao, M. Zheng, A. Choudhuri, T. Chen, C. Chen, and Z. Wu. 3d vision-language gaussian splatting. *arXiv preprint arXiv:2410.07577*, 2024.
- [36] S. Peng, K. Genova, C. Jiang, A. Tagliasacchi, M. Pollefeys, T. Funkhouser, et al. Openscene: 3d scene understanding with open vocabularies. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 815–824, 2023.
- [37] Y. Peng, H. Wang, Y. Liu, C. Wen, Z. Dong, and B. Yang. Gags: Granularity-aware feature distillation for language gaussian splatting. *arXiv preprint arXiv:2412.13654*, 2024.
- [38] C. R. Qi, H. Su, K. Mo, and L. J. Guibas. Pointnet: Deep learning on point sets for 3d classification and segmentation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 652–660, 2017.
- [39] C. R. Qi, L. Yi, H. Su, and L. J. Guibas. Pointnet++: Deep hierarchical feature learning on point sets in a metric space. *Advances in neural information processing systems*, 30, 2017.
- [40] 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*, pages 20051–20060, 2024.
- [41] J. Qiu, L. Liu, Z. Su, and T. Lin. Gls: Geometry-aware 3d language gaussian splatting. *arXiv preprint arXiv:2411.18066*, 2024.
- [42] A. Radford, J. W. Kim, C. Hallacy, A. Ramesh, G. Goh, S. Agarwal, G. Sastry, A. Askell, P. Mishkin, J. Clark, et al. Learning transferable visual models from natural language supervision. In *International conference on machine learning*, pages 8748–8763. PMLR, 2021.
- [43] D. Rozenberszki, O. Litany, and A. Dai. Unscene3d: Unsupervised 3d instance segmentation for indoor scenes. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 19957–19967, 2024.
- [44] W. Shen, G. Yang, A. Yu, J. Wong, L. P. Kaelbling, and P. Isola. Distilled feature fields enable few-shot language-guided manipulation. *arXiv preprint arXiv:2308.07931*, 2023.
- [45] J.-C. Shi, M. Wang, H.-B. Duan, and S.-H. Guan. Language embedded 3d gaussians for open-vocabulary scene understanding. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 5333–5343, 2024.
- [46] Y. Siddiqui, L. Porzi, S. R. Buló, N. Müller, M. Nießner, A. Dai, and P. Kontschieder. Panoptic lifting for 3d scene understanding with neural fields. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 9043–9052, 2023.
- [47] J. Straub, T. Whelan, L. Ma, Y. Chen, E. Wijmans, S. Green, J. J. Engel, R. Mur-Artal, C. Ren, S. Verma, et al. The replica dataset: A digital replica of indoor spaces. *arXiv preprint arXiv:1906.05797*, 2019.
- [48] L. Tang, Y. Zhan, Z. Chen, B. Yu, and D. Tao. Contrastive boundary learning for point cloud segmentation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 8489–8499, 2022.
- [49] B. Wang, L. Chen, and B. Yang. Dm-nerf: 3d scene geometry decomposition and manipulation from 2d images. *arXiv preprint arXiv:2208.07227*, 2022.
- [50] S. Wang, V. Leroy, Y. Cabon, B. Chidlovskii, and J. Revaud. Dust3r: Geometric 3d vision made easy. *arXiv preprint arXiv:2312.14132*, 2023.
- [51] Y. Wang, X. Wei, M. Lu, and G. Kang. Plgs: Robust panoptic lifting with 3d gaussian

- splatting. *arXiv preprint arXiv:2410.17505*, 2024.
- [52] X. Wei, R. Zhang, J. Wu, J. Liu, M. Lu, Y. Guo, and S. Zhang. Nto3d: Neural target object 3d reconstruction with segment anything. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 20352–20362, 2024.
  - [53] Y. Wu, J. Meng, H. Li, C. Wu, Y. Shi, X. Cheng, C. Zhao, H. Feng, E. Ding, J. Wang, et al. Opengaussian: Towards point-level 3d gaussian-based open vocabulary understanding. *arXiv preprint arXiv:2406.02058*, 2024.
  - [54] Z. Xia, Y. Liu, X. Li, X. Zhu, Y. Ma, Y. Li, Y. Hou, and Y. Qiao. Scpnet: Semantic scene completion on point cloud. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 17642–17651, 2023.
  - [55] X. Xu, H. Chen, L. Zhao, Z. Wang, J. Zhou, and J. Lu. Embodiedsam: Online segment any 3d thing in real time. *arXiv preprint arXiv:2408.11811*, 2024.
  - [56] M. Ye, M. Danelljan, F. Yu, and L. Ke. Gaussian grouping: Segment and edit anything in 3d scenes. *arXiv preprint arXiv:2312.00732*, 2023.
  - [57] F. G. Zanjani, H. Cai, H. Ackermann, L. Mirvakhabova, and F. Porikli. Planar gaussian splatting. *arXiv preprint arXiv:2412.01931*, 2024.
  - [58] W. Zhang, L. Zhang, P. Hu, L. Ma, Y. Zhuge, and H. Lu. Bootstrapping clustering of gaussians for view-consistent 3d scene understanding. *arXiv preprint arXiv:2411.19551*, 2024.
  - [59] S. Zhi, T. Laidlow, S. Leutenegger, and A. J. Davison. In-place scene labelling and understanding with implicit scene representation. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 15838–15847, 2021.
  - [60] 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*, pages 21676–21685, 2024.