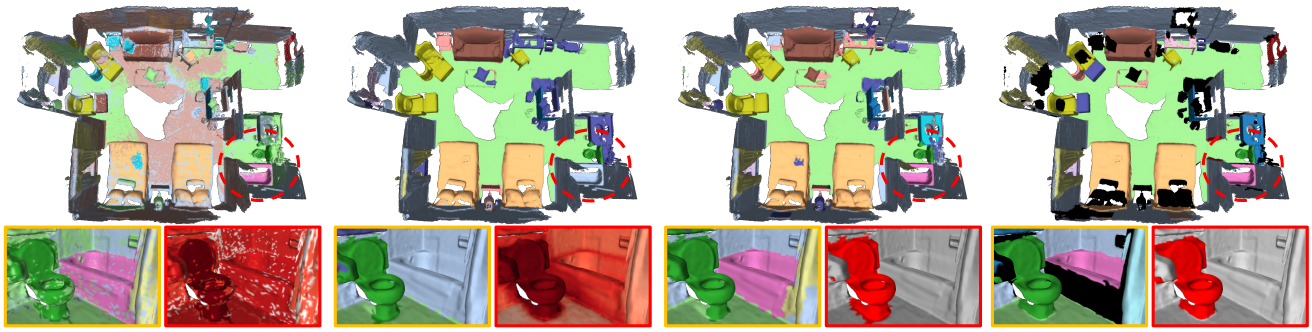
PanoGS: Gaussian-based Panoptic Segmentation for 3D Open Vocabulary Scene Understanding

Hongjia Zhai[1] Hai Li[2] Zhenzhe Li[1] Xiaokun Pan[1] Yijia He[2] Guofeng Zhang[1†]
[1]State Key Lab of CAD & CG, Zhejiang University [2]RayNeo



OpenGaussian [45] OpenScene [30] **Ours** Ground Truth

Figure 1. **Open Vocabulary 3D Panoptic Scene Understanding**. Visualization of open-vocabulary semantic segmentation (yellow boxes) and object query with text *toilet* (red boxes). Our PanoGS can achieve more accurate segmentation results and generate 3D instance-level results for open-vocabulary text queries, unlike previous methods that generate heatmaps between scene features and text queries.

Abstract

Recently, 3D Gaussian Splatting (3DGS) has shown encouraging performance for open vocabulary scene understanding tasks. However, previous methods cannot distinguish 3D instance-level information, which usually predicts a heatmap between the scene feature and text query. In this paper, we propose PanoGS, a novel and effective 3D panoptic open vocabulary scene understanding approach. Technically, to learn accurate 3D language features that can scale to large indoor scenarios, we adopt the pyramid tri-plane to model the latent continuous parametric feature space and use a 3D feature decoder to regress the multi-view fused 2D feature cloud. Besides, we propose language-guided graph cuts that synergistically leverage reconstructed geometry and learned language cues to group 3D Gaussian primitives into a set of super-primitives. To obtain 3D consistent instance, we perform graph clustering based segmentation with SAM-guided edge affinity computation between different super-primitives. Extensive experiments on widely used datasets show better or more competitive performance on 3D panoptic open vocabulary scene understanding. Project page: https://zju3dv.github.io/panogs.

1. Introduction

3D scene understanding is a critical problem in computer vision that enables humans or intelligent agents to comprehensively understand the 3D scenes and facilitate the downstream applications [15, 28, 49, 55, 58, 59]. They usually incorporate vision-language models (VLMs) [2, 21, 33] for a fine-grained and holistic understanding of the environment.

Recently, Neural Radiance Fields (NeRF) [27] and 3D Gaussian Splatting (3DGS) [16] have rapidly gained much research attention for novel view synthesis. Due to the fast rendering ability and explicit scene representation of 3DGS, it has been widely integrated into reconstruction [4, 14, 54], generation [39, 52], and understanding [51, 60, 63]. While the domain of combining 3DGS with scene understanding has recently made several progress [1, 13, 32, 34, 63], these approaches are primarily designed for 2D pixel-level semantic segmentation with rendered 2D language feature maps, which cannot distinguish the different objects with the same semantics in the 3D space.

Although previous 3DGS-based approaches [1, 32, 34, 45] have achieved impressive performance that combines VLMs with 3DGS for open vocabulary scene understanding, there also exist the following limitations that prevent 3DGS-based approaches for panoptic open vocabulary scene understanding: *1) inaccurate 3D language feature learning*. The discrete features attached to the Gaussian

†Corresponding author.

primitive can affect the inherent smoothness of language features for semantically similar objects [32, 34, 60]. The alpha-blending technique accumulates the 3D discrete features of primitives based on the opacity weight, which leads to a domain gap between 2D and 3D feature space [45, 57]. And the 2D feature compression [32] and feature distillation [34, 63] will inevitably damage the distinguishing ability of learned language features. *2) unable to recognize 3D instance-level information*. Previous methods [30, 45] usually predict a heatmap of similarity between the language feature of the 3D scene and the text query. These approaches may lead to inconsistent instance segmentation results and can not distinguish the multi-instance objects of the same semantics. However, instance-level information is essential for 3D panoptic scene understanding.

To address the aforementioned two limitations, we propose PanoGS, a novel and effective 3DGS-based approach for 3D panoptic open vocabulary scene understanding. Firstly, to enhance the representation ability and spatial smoothness of our learned language feature, we use a pyramid tri-plane to model the latent continuous parametric feature space of the 3D scene. We use the 2D multi-view fused feature cloud and confidence to perform distillation of learned language features in 3D space, not 2D rendered space, which can avoid the domain gap between 2D and 3D feature spaces caused by alpha-blending. In addition, to obtain 3D instance information, we formulate 3D instance segmentation as a graph clustering problem. To build the 3D scene graph, we use our language-guided graph cuts algorithm to group Gaussian primitives into geometrically and semantically consistent graph vertices, that is super-primitives. Besides, we use the 2D segmentation model SAM [19] to obtain 2D mask labels and construct the affinity between 3D super-primitives based on the mult-view consistency of 2D mask label distribution. Finally, a progressive clustering strategy is used to obtain globally consistent instance information. Overall, the technical contributions of our approach are summarized as follows:

- We propose PanoGS, the first 3DGS-based approach for 3D panoptic open vocabulary scene understanding.
- We learn an inherent smooth and accurate 3D language feature field based on our latent pyramid tri-plane, which is optimized by 2D fused feature cloud and confidence.
- We design an effective graph clustering based segmentation algorithm to synergistically leverage geometric and semantic cues to obtain consistent 3D instances.
- We conduct extensive experiments on commonly used datasets to demonstrate the 3D panoptic segmentation performance of our approach.

2. Related Work

Panoptic Segmentation. Kirillov *et al*. [18] propose the 2D panoptic segmentation task as a new research topic. While many methods [5, 6, 31] focus on improving the reasoning ability of CNN models to understand individual images, there is still a gap in 3D panoptic scene understanding due to the lack of 3D training data. To enhance the 3D panoptic segmentation, some works aim to lift 2D panoptic predictions into 3D scene space, with different scene representations, such as point cloud [11, 61], voxels [29], 3DGS [44], and implicit representation [10, 20, 35, 62]. However, those works are rather limited to close-set panoptic segmentation and can not recognize the objects of unseen classes.

3D Gaussian Splatting. Recently, 3DGS [16] has demonstrated remarkable advancements in many tasks of 3D computer vision [4, 14, 39, 57]. Compared to previous NeRF-based approaches [22, 23, 27, 56], 3DGS represents the scene with a set of anisotropic 3D Gaussian primitives explicitly. To enforce geometric consistency, some studies aim to control the shape of the primitives [14], use unbiased depth rendering [4], and introduce geometric regularization during the optimization process, such as monocular depth [25, 47], and normals [41, 46]. Besides, some works extend 3DGS to model dynamic scenes with deformable fields [50] and explicit motion estimation [36, 43]. Meanwhile, some work equips 3DGS with scene understanding by extending each primitive with learnable language embeddings for open-vocabulary 3D queries. LangSplat [32] uses an auto-encoder to compress the dimension of language features. N2F2 [1] uses a tri-plane [3] as the additional feature encoding to reduce parameters.

3D Open Vocabulary Scene Understanding. Inspired by the successful 2D visual-language models (VLMs) [2, 21, 33], some approaches aim to learn 3D consistent feature fields to model the semantic property with explicit 3D structures [30] (*e.g*., point clouds). Besides, some researchers try to perform scene understanding with neural implicit representations [16, 27]. To achieve this, LERF [17] and N3F [40] are early exploratory works, which optimize an additional field branch to align the feature space with VLMs (*e.g*., CLIP [33], DINO [2]). Recent efforts [32, 34, 60, 63] have combined 3DGS with 2D scene understanding techniques due to its advantage of explicit representation. However, they cannot distinguish the different objects with the same semantics, which is not suitable for panoptic scene understanding.

3. Method

As shown in Fig. 2, given multi-view posed images $\{I_i, D_i\}_{i=1}^{m}$, we can perform 3D open vocabulary scene understanding. To achieve this goal, we propose an effective 3D language feature field learning module, which adopts a pyramid tri-plane to model the latent continuous parametric feature space and regress the language feature from fused feature cloud and confidence (Sec. 3.2). Besides, we apply a graph clustering based segmentation algorithm to obtain 3D

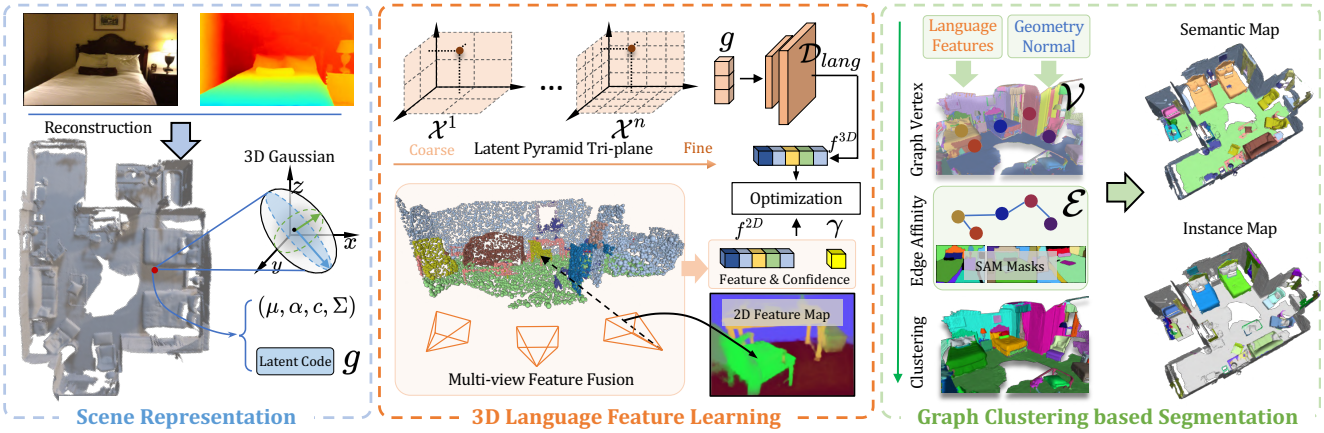


Figure 2. Overview of our approach. (a) Given posed RGB-D images, we reconstruct the scene with 3D Gaussian primitives, and each primitive is associated with additional latent language code g generated from a latent continuous pyramid tri-plane feature space. (b) After the geometry reconstruction, we obtain 2D fused primitive-level features and confidences via back projection, which is used for efficient 3D language feature regression and latent pyramid tri-plane and 3D decoder optimization. (c) We perform a language-guided graph cuts algorithm to construct super-primitive and use the 2D instance mask generated by SAM [19] to conduct progressive graph clustering.

consistent instances based on the scene graph reconstructed with geometry and language cues (Sec. 3.3).

3.1. Scene Representation

Benefiting from the efficiency of 3DGS [16], we take it as our scene representation for 3D panoptic scene understanding. Following the fast differentiable rasterization [26], we can render the 3D scene properties to the 2D image plane.

$$\hat{A} = \sum_i a_i \cdot \alpha_i \cdot \prod_{j=1}^{i-1}(1-\alpha_j), \tag{1}$$

where $\hat{A}$ are the rendered 2D scene information (*e.g.*, color, depth). a_i and α_i denote the 3D property and opacity contribution of i-th Gaussian primitive, and $\prod_{j=1}^{i-1}(1-\alpha_j)$ is the accumulated transmittance.

Following previous works [16, 25], the appearance and geometry loss functions are used for optimization:

$$\mathcal{L}_{recon} = w_1 \cdot \mathcal{L}_c(\hat{I}, I) + w_2 \cdot \mathcal{L}_d(\hat{D}, D), \tag{2}$$

where $\{w_i\}$ are the weights for each optimization component and $\mathcal{L}_c$ and $\mathcal{L}_d$ are the L1 loss terms between rendered color/depth and input color/depth.

Similar to previous 3DGS-based scene understanding works [32, 34, 45, 60, 63], we additionally attach a latent low-dimensional code g for each Gaussian primitive. Different from them, to reduce the memory requirement and spatial noise introduced by the discrete representation, we don't explicitly save the individual features but model it via a latent continuous pyramid tri-plane feature space. This design allows us to learn better scene language feature fields for panoptic 3D scene understanding.

3.2. 3D Language Feature Learning

As an explicit modeling method, 3DGS can not store high-dimensional language features for each primitive. Previous approaches [32, 34, 63] apply quantization or compression to reduce the dimensions. However, these operations inevitably reduce the accuracy and distinguishability of the learned language features. To learn accurate 3D language features, we regress the feature via decoding the latent language code sampled from the 3D pyramid tri-plane.

Latent Pyramid Tri-plane. As shown in the middle part of Fig. 2, we adopt a pyramid tri-plane to model the latent continuous parametric feature space of the 3D scene. Compared with the previous discrete feature of each Gaussian primitive, our method can directly regress the language features of the 3D scene, which is not affected by the bias introduced by the alpha-blending as pointed out in [45, 57]. Specifically, given a 3D position μ, we first query its multi-resolution latent language code via the following equation:

$$g(\mu) = \sum_i^n \{\mathcal{T}(\mu, \mathcal{X}_{xy}^i), \mathcal{T}(\mu, \mathcal{X}_{yz}^i), \mathcal{T}(\mu, \mathcal{X}_{xz}^i)\}, \tag{3}$$

where $\mathcal{T}(\cdot)$ is the trilinear interpolation operation, and $\mathcal{X}_{xy}^i$, $\mathcal{X}_{yz}^i$, $\mathcal{X}_{xz}^i$ represent the decomposed feature planes of i-th resolution level in the pyramid.

Due to memory constraints, the dimension of the latent code $g(\mu)$ is usually much less than the original language feature. So, to obtain the high-dimensional language feature for open vocabulary scene understanding, we use the 3D language feature decoder $\mathcal{D}_{lang}$, which transforms the low-dimensional latent code into high-dimensional lan-

guage feature with the following equation:

$$f^{3D}(\mu) = \mathcal{D}_{lang}(g(\mu)), \tag{4}$$

where $f^{3D}(\mu) \in \mathbb{R}^{D_l}$ is the decoded high-dimensional language and D_l is the dimension of the language feature.

Multi-view Feature Fusion. After obtaining the reconstructed Gaussian primitives, we can project primitives into multi-view 2D images. We use LSeg [21] to extract visual-language feature maps for RGB images. So, for a 3D Gaussian primitive p_i, we can obtain its multi-view 2D feature vectors $\{f_1, \cdots, f_m\}$ from m 2D feature maps. To obtain its fused primitive-level feature f_i^{2D}, we adopt the weighted average on the multi-view 2D feature vectors according to the occlusion and observation of the 3DGS primitive, $f_i^{2D} = \Phi(\{f_1, \cdots, f_m\})$ and $\Phi(\cdot)$ is the pooling operation. Additionally, the fused 3D primitive-level features may have different confidence due to multi-view inconsistency and occlusion. To measure this, we compute the confidence value γ_i^{2D} for i-th primitive p_i as follows:

$$\gamma_i^{2D} = \frac{\mathit{Obs}(p_i)}{\sum_{D_l} \mathit{Var}(\{f_1, f_2, \cdots, f_m\})}, \tag{5}$$

where $\mathit{Obs}(p_i)$ count the normalized number of valid observations of primitive p_i, $\mathit{Var}(\cdot)$ denote the variance of the observed multi-view language features $\{f_1, f_2, \cdots, f_m\}$, and D_l is the dimension of the observed language feature f_i.

So, for the reconstructed scene with N Gaussian primitives, we can obtain the primitive-level 2D fused feature cloud $\{f_i^{2D}\}_{i=1}^N$ and confidence $\{\gamma_i^{2D}\}_{i=1}^N$.

Language Feature Distillation. With the fused features cloud and confidence, we can optimize our 3D latent pyramid tri-plane $\{\mathcal{X}_{xy}^i, \mathcal{X}_{yz}^i, \mathcal{X}_{xz}^i\}$ and 3D language feature decoder $\mathcal{D}_{lang}$ to learn accurate feature representation. Specifically, the latent codes g from the pyramid tri-plane are assigned to 3D Gaussian primitives, and the 3D language features of primitives are decoded from their latent codes. So, for i-th primitive, its 2D fused language feature f_i^{2D} and 3D learned feature f_i^{3D}, we can use the following equation for optimization:

$$\mathcal{L}_{feat} = \sum_i^N \gamma_i^{2D} \cdot |1 - \cos(\mathcal{D}_{lang}(g_i), f_i^{2D})|, \tag{6}$$

where $\cos(\cdot)$ denotes the cosine similarity function.

3.3. Graph Clustering based Segmentation

Previous methods [35, 62] lift 2D information to 3D space for feature learning, which may lead to 3D multi-view inconsistencies. Different from them, we directly cluster the reconstructed 3D Gaussian primitives into several disjoint subsets, where each subset can represent a class-agnostic instance in the 3D scene. So, to achieve this goal, we formulate this problem as a graph clustering task and construct the 3D scene graph $\mathcal{G} = (\{\mathcal{V}_i\}, \{\mathcal{E}_{ij}\})$ based on our reconstructed Gaussian primitives and learned 3D language feature in Sec. 3.2. In the following, we elaborate on the details of how to construct the graph vertex $\{\mathcal{V}_i\}$, edge affinity $\{\mathcal{E}_i\}$, and progressive graph clustering.

Graph Vertex Construction. Viewing each 3D Gaussian primitive as a vertex and building a fully connected edge weight graph between all vertexes is impractical for solving graph clustering problems due to indoor scenes usually containing millions of 3D Gaussian primitives. Previous works [12, 48, 53] use normal information and graph cuts algorithm [9] to group 3D points into a set of superpoints. However, only using the geometric properties can lead to over-segmentation or under-segmentation. Benefiting from our language feature learned in Sec. 3.2, we can simultaneously take the local geometry information and global semantic information into consideration for grouping individual Gaussian primitives into a set of super-primitives.

To obtain geometrically and semantically consistent super-primitives $\{\mathcal{V}_i\}$, we perform the language-guided graph cuts with our language feature during the merge process. Additionally, to access the global semantic property of super-primitives, we retrieve the language feature of the current super-primitive, which is updated during the merge process. In language-guided graph cuts, to judge whether two super-primitives merge into a new vertex, we adopt the following criteria:

$$\Delta = \mathbb{1}\left((n_i \cdot n_j) > \lambda_n\right) \cdot \mathbb{1}\left((f_i^{3D} \cdot f_j^{3D}) > \lambda_f\right), \tag{7}$$

where $\mathbb{1}(\cdot)$ is the indication function that equals 1 when the condition is satisfied, n_i and f_i^{3D} are the normal and language features within the current super-primitive. λ_n and λ_f are the threshold parameters at this iteration in our language-guided graph cuts process.

With our language-guided graph cuts algorithm, we can obtain geometrically and semantically consistent super-primitives rather than only using geometry information. After traversing all reconstructed Gaussian primitives, we view each super-primitive as a vertex $\mathcal{V}_i$ in the graph $\mathcal{G}$, which is represented by the disjoint sets of super-primitives.

Edge Affinity Computation. After obtaining graph vertices $\{\mathcal{V}_i\}$, we build edge and compute affinity based on the spatial adjacency relationships between these super-primitives. Inspired by recent works [38, 45, 48, 51, 53] that use powerful segmentation models [19, 24] to generate multi-view 2D instance masks as the guidance of instance-level feature learning and clustering. Similarly, we also use the 2D multi-view instance masks generated by SAM [19] to aid the edge affinity computation.

To build the affinity between two super-primitives $\mathcal{V}_i$ and $\mathcal{V}_j$, we first project them into k-th image mask according to the camera intrinsic. The 3D Gaussian primitives inside $\mathcal{V}_i$ will fall in k-th instance mask, and each primitive should

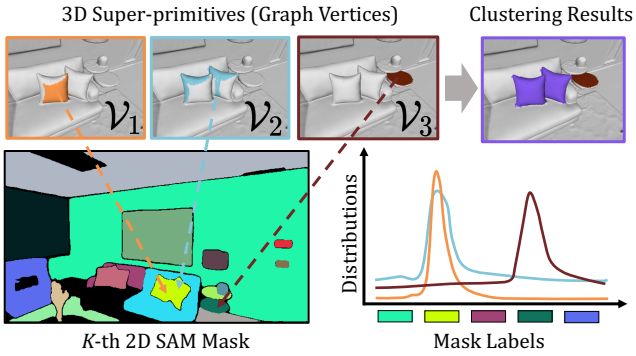


Figure 3. By projecting primitives inside different graph vertices into k-th 2D SAM mask, $\mathcal{V}_1$ and $\mathcal{V}_2$ have similar mask label distributions, while $\mathcal{V}_3$ has different mask label distributions. So, we can cluster $\mathcal{V}_1$ and $\mathcal{V}_2$ into the same category based on the distance between the distributions.

have a 2D mask label in the k-th mask. So, we can obtain its 2D mask label distribution q_i based on the segmentation results of SAM [19]. As shown in Fig. 3, two super-primitives that belong to the same 2D instance should have similar 2D mask label distributions. So, we can take the 2D mask label distribution as our 3D super-primitives consistency criterion. To measure the difference between the 2D mask label distribution of two 3D super-primitives, we use the Jensen-Shannon divergence (JSD):

$$\mathcal{E}_{ij}^{k} = \frac{1}{2}\sum_{z}[q_i(z)\log(\frac{q_i(z)}{y(z)}) + q_j(z)\log(\frac{q_j(z)}{y(z)})], \quad (8)$$

where $y = (q_i + q_j)/2$ is the average of two mask label distributions.

Evidently, affinity from a single view may be noisy and inaccurate. We should consider the multi-view consistency when performing 3D consistent panoptic segmentation. So, we aggregate multi-view affinity to achieve cross-view consistency based on the proportion of primitives that can be observed in different views. The final multi-view affinity is defined in the following equation:

$$\mathcal{E}_{ij} = \frac{1}{k}\sum_{k=1}(\frac{\texttt{Vis}(\mathcal{V}_i)}{|\mathcal{V}_i|}\cdot\frac{\texttt{Vis}(\mathcal{V}_j)}{|\mathcal{V}_j|}\cdot\mathcal{E}_{ij}^{k}) \quad (9)$$

where $\texttt{Vis}(\cdot)$ function indicates the number of visible primitives in the current viewpoint and $|\mathcal{V}_i|$ denotes the number of primitives insides $\mathcal{V}_i$.

When we obtain the multi-view consistency criterion of two super-primitives, we can merge them into the same instance group based on the following progressive clustering process.

Progressive Graph Clustering. According to the graph $\mathcal{G} = (\{\mathcal{V}_i\}, \{\mathcal{E}_{ij}\})$ built in the previous section, we perform graph clustering that merges $\{\mathcal{V}_i\}$ with large affinity scores into the same instance. To obtain the global consistent clustering results, we adopt a progressive local-to-global way to merge super-primitives with spatial neighbor connections. Specifically, we first cluster the local super-primitives with high-affinity scores and merge them into large super-primitives. During each iteration, we update the graph vertexes and edge affinity for the next iteration due to the changes in the scene graph structure. We update the affinity threshold during progressive clustering, which is linearly reduced from 0.9 to 0.6 with 4 iterations.

3.4. Open Vocabulary Panoptic Segmentation

When we finish progressive graph clustering, we can obtain a set of non-overlapping clustering groups where each entry represents a 3D class-agnostic instance. Besides, with the 3D feature decoder optimized in Sec. 3.2, we can obtain the 3D language feature for each primitive. For 3D open vocabulary panoptic segmentation, we assume that the Gaussian primitives inside the same super-primitive should belong to the same semantic category. Therefore, we use a prediction voting method to calculate the semantic category of the super-primitive, which can get more complete instance-level semantic segmentation results.

4. Experiments

In this section, we first describe our experimental setting and then present quantitative and qualitative results of our approach and state-of-the-art baselines on two commonly used datasets. Additionally, we perform a detailed ablation study to justify our design choices.

4.1. Experimental Settings

Datasets. Following the previous works [30, 45], we evaluate our method on two widely used indoor scene datasets, Replica [37] and ScanNetV2 [8] for both quantitative and qualitative evaluations. The Replica dataset contains high-quality indoor scenes with carefully annotated ground-truth semantic and instance labels. For a fair comparison, we take the commonly-used 8 scenes $\{\texttt{room0-2,office0-4}\}$ for evaluation. The ScanNetV2 [8] dataset consists various of challenging indoor scenes with different numbers of RGB-D frames for each sequence, as well as reconstructed point clouds and GT 3D point-level semantic labels. Following [45], we use the same 10 selected sequences and settings for the evaluation of scene understanding.

Evaluation Metrics. For the evaluation of open vocabulary 3D scene understanding, we evaluate the performance of our method on four widely-used metrics: point cloud mean Intersection over Union (mIoU), mean Accuracy (mAcc.), and 3D Panoptic Reconstruction Quality (PRQ) [7] which is modified from the common 2D panoptic segmentation quality [18]. In our experiments, we use the *thing*-level metric, PRQ (T), and *stuff*-level metric, PRQ (S) for evaluation.

Method	mIoU	mAcc.	PRQ (T)	PRQ (S)
Open-vocab. semantic + Sup. mask [42]				
LangSplat [32]	3.78	9.11	—	—
LangSplat* [32]	29.47	45.29	22.57	28.44
OpenGaussian [45]	24.73	41.54	—	—
OpenGaussian† [45]	24.89	37.35	22.87	19.71
OpenScene(*Dis.*) [30]	46.91	68.50	<u>43.77</u>	<u>40.69</u>
OpenScene(*Ens.*) [30]	<u>47.63</u>	<u>69.74</u>	43.53	40.43
Ours	**50.72**	**70.20**	33.84	36.22
Ours + *Sup. mask [42]*	**50.72**	**70.20**	**49.26**	**48.24**

Table 1. 3D semantic and panoptic segmentation results on ScanNetV2 [8]. The results of [32] and [45] are taken from [45] and OpenScene [30] are obtained from their pre-trained model. $*$ indicates our better implementation. † indicates no Gaussian filter is used for the evaluation of panoptic segmentation.

Method	mIoU	mAcc.	PRQ (T)	PRQ (S)
Open-vocab. semantic + Sup. mask [42]				
LangSplat* [32]	4.82	10.03	8.29	1.28
OpenScene(*Dis.*) [30]	44.32	56.14	31.43	10.95
OpenScene(*Ens.*) [30]	<u>49.03</u>	<u>62.89</u>	33.04	<u>11.84</u>
Ours	**54.98**	**67.35**	**43.04**	**30.60**
Ours + *Sup. mask [42]*	**54.98**	**67.35**	<u>40.80</u>	11.31

Table 2. 3D semantic and panoptic segmentation results on Replica [37]. The results of OpenScene [30] are obtained from their pre-trained model. $*$ indicates our better implementation of LangSplat [32].

Implementation Details. Following [30, 45], we use LSeg [21] as our pixel-aligned visual-language feature extractors for ScanNetV2 and Replica datasets. To extract the language feature of the text query, we use the OpenCLIP [33] ViT-B/16 model. For the 2D masks, we use the ViT-H SAM [19] model to segment images. More details are provided in our supplementary material.

Baselines. We compare our approach with recent 3DGS-based approaches, LangSplat [32], OpenGaussians [45] and point-cloud based method, OpenScene [30]. Due to the performance of LangSplat [32] reported in [45] is too worse, we modify it for better performance, which is indicated by $*$. And † indicates that no Gaussian filtering is used in [45] for the evaluation of panoptic segmentation. Besides, due to these methods can only extract the point-level language features, we use the fully supervised 3D instance segmentation approach SoftGroup [42] (trained on ScanNetV2 [8]) to provide instance mask proposals for the comparison of panoptic segmentation.

4.2. Main Experiments

We evaluate the 3D panoptic segmentation metrics of our approach and baseline on two commonly used ScanNetV2 [8] and Replica [37] datasets. Due to the open-source codes of OpenGaussian [45] is not complete, we can't obtain its 3D scene understanding performance on the Replica dataset. For OpenScene [30] , we use its 3D Distill (*Dis.*) and 2D/3D Ensemble (*Ens.*) variants for comparison. The best results are shown in **bolded**, and the second-best results are represented in <u>underlined</u>.

3D Semantic Segmentation. The averaged quantitative semantic segmentation results are shown in Tab. 1 and Tab. 2, respectively. According to the 3D semantic segmentation results in the table, our approach achieves the best results on the mIoU and mAcc metrics of two datasets. Compared with 3DGS-based approaches [32, 45], our method can achieve significant improvements. Benefiting from learning language features in the 3D space, we can learn consistent features for each Gaussian primitive, and avoid the bias introduced by alpha-blending. Previous 3DGS-based approaches learn individual and discrete features for each Gaussian primitive which can lead to spatial noise and destroy the smoothness of the semantic features. Our approach inherently learns language features from a 3D latent pyramid tri-plane. Also, OpenScene [30] which utilizes the 3D and 2D information to perform scene understanding has achieved better performance than current 3DGS-based methods. Our method still performs better than OpenScene on the open vocabulary 3D semantic segmentation task. The qualitative semantic segmentation results with open vocabulary query of ScanNetV2 [8] are shown in Fig. 4. Our method can achieve consistent segmentation results, while previous 3DGS-based methods often lead to noisy segmentation results, as shown in the first two columns of Fig. 4. In addition, compared with OpenScene, we can achieve better segmentation results on the long-tail categories.

3D Panoptic Segmentation. The averaged quantitative results are also shown in Tab. 1 and Tab. 2, respectively. For the 3D panoptic segmentation performance, due to the compared baselines only output point-level segmentation results, we use the fully supervised 3D instance segmentation approach, SoftGroup [42] (trained on ScanNetV2), to generate 3D instance proposals for them. Since SoftGroup [42] has a better instance segmentation performance on ScanNetV2, our PRQ (T) and PRQ (S) are slightly worse than the combination of OpenScene+SoftGroup. However, when we use the 3D instance maks generated by SoftGroup, we achieve the best results on 3D panoptic reconstruction quality. For the Replica dataset, the segmentation results of SoftGroup are worse than our approach. Based on our learned language features and clustering-based segmentation results, we can achieve the best results in terms of PRQ (T) and PRQ (S). Besides, we also show the qualitative panoptic segmentation results of SoftGroup [42] and ours in Fig. 5. As can be seen from the figure, we can generalize better than SoftGroup [42] for the instance segmentation

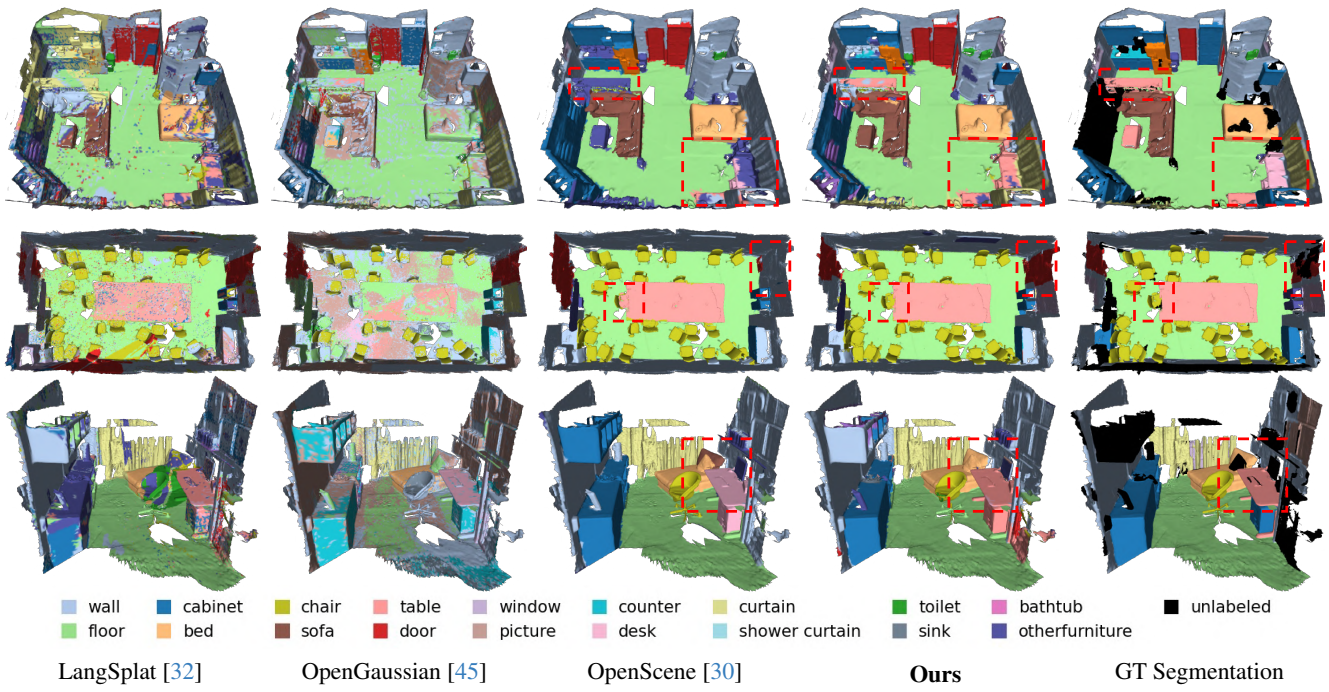


Figure 4. Qualitative 3D semantic segmentation comparison of ScanNetV2 [8]. Our approach outperforms recent 3DGS-based approaches, LangSplat [32] and OpenGaussian [45], by a large margin. Compared with OpenScene [30], we can achieve better segmentation results on thing-level objects.

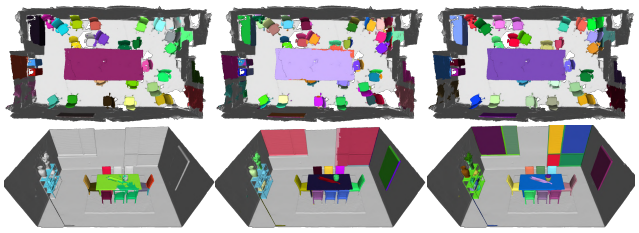


Figure 5. Qualitative 3D panoptic segmentation comparison. We show two reconstructed panoptic maps selected from ScanNetV2 [8] and Replica [37] datasets.

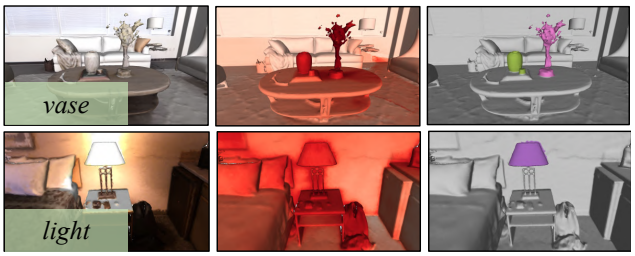


Figure 6. Qualitative results of open vocabulary query. The query text is in the lower left of the RGB image. For OpenScene, the redder the color, the higher the similarity. We use different colors to distinguish different instances found in the query.

results on the Replica [37] dataset.

Open Vocabulary Query. We show the qualitative open vocabulary query results of two selected scenes in Fig. 6. Previous methods [30, 32, 34, 45] calculated the similarity between scene features and text queries, and they can not distinguish different objects with the same semantics. However, our approach can obtain instance-level information of different objects with the same semantics through our clustering (shown in the first row of Fig. 6).

4.3. Ablation Studies and Analysis

We conduct ablation studies to analyze the effectiveness of 3D language feature learning and graph clustering based segmentation modules. When we analyze one module, the other one is fixed and under the default setting.

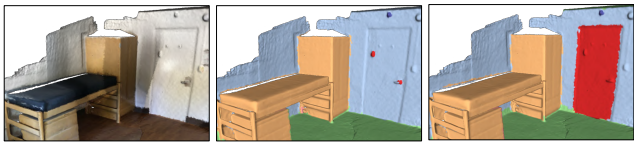
3D Dec.	*Py. Tri.*	mIoU	mAcc.	PRQ (T)	PRQ (S)
✗	✗	24.51	40.48	20.57	23.07
✓	✗	36.39	53.17	23.68	33.59
✓	✓	**50.72**	**70.20**	**33.84**	**36.22**

Table 3. Ablation studies of our 3D language feature learning module. The results are evaluated on the ScanNetV2 [8] dataset.

Effects of the designs for our 3D language feature learn-

JSD.	*Lang.*	*Vot.*	mIoU	mAcc.	PRQ (T)	PRQ (S)
✗	✗	✗	50.21	61.69	20.28	17.42
✓	✗	✗	50.21	61.69	33.22	25.21
✓	✓	✗	50.21	61.69	35.28	27.41
✓	✗	✓	52.69	63.25	40.10	**31.57**
✓	✓	✓	**54.98**	**67.35**	**43.04**	30.60

Table 4. Ablation studies of our graph clustering based segmentation module. The results are evaluated on the Replica [37] dataset.



RGB w/o *Lang.* w *Lang.*

Figure 8. Qualitative comparison of using language-guided graph cut for graph vertex construction. Using our learned language feature can distinguish the door (shown in red color).

ing module. In Tab. 3, we show the quantitative analysis of our feature learning module. *3D Dec.* and *Py. Tir.* indicate using the 3D decoder to regress language features from the projected multi-view primitive-level feature and using our latent pyramid tri-planes, respectively. w/o *3D Dec.* and *Py. Tri.* means that we use a 2D autoencoder similar to LangSplat [32] and replace triplane with positional encoding for feature learning. As shown in the table, compared with directly performing distillation with 2D auto encode-decoder (without *3D Dec.* and *Py. Tri.*), using 3D feature distillation and latent parametric encoding both can lead to better performance and reduce the noise introduced by the discrete Gaussian language feature. Additionally, on the left of Fig. 7, we also show the language feature learning gap of using 2D and 3D distillation way. From the figure, we can know that for large scenes, using the 2D distillation, the similarity of the learned language feature can only reach 0.9. But using the coordinate-based 3D distillation approach and our latent pyramid tri-plane, the feature learning performance can reach 0.95 and close to 1, respectively. Besides, the performance of using the confidence during distillation is shown in the right part of Fig. 7. The results also validate the effectiveness of our confidence based feature learning, which can efficiently reduce the impact of features from unobserved and unreliable areas.

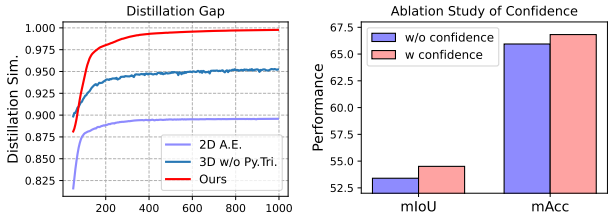


Figure 7. Ablation studies of our 3D language learning module. The results are evaluated on Replica [37] dataset.

Effects of the designs for our graph clustering based segmentation module. In Tab. 4, we show the quantitative analysis of our graph clustering based segmentation module. *JSD.*, *Lang.*, and *Vot.* indicate using multi-view JSD of mask label distributions to construct graph edge affinity, using language-guided graph cuts to construct graph vertices, and using predictions voting (Sec. 3.4) for semantic segmentation, respectively. Besides, w/o *JSD.* means that we use the similarity of learned language features between super-primitives to construct edge affinity. As can be seen from the table, compared with using language features (w/o *JSD.*) to build graph edge affinity, using our multi-view affinity from mask label distributions will significantly improve our 3D panoptic reconstruction quality (PRQ (T): 33.22 *v.s* 20.28 and PRQ (S): 25.21 *v.s* 17.42). Using our language-guided graph cuts can avoid merging different semantic objects with similar structures into the same vertex/instance, such as wall and door, as shown in Fig. 8. It also can further improve the panoptic segmentation performance of 3D instances. In addition, after obtaining accurate clustering results, when performing semantic segmentation, we vote the prediction results inside the same super-primitive to obtain consistent prediction results for objects. The results validate *Vot.* is also effective. It can ensure that the primitives inside a 3D instance object have consistent semantic prediction results, which can increase instance-level IoU metrics for the predicted results and ground truth in 3D panoptic segmentation.

5. Conclusion

In this paper, we propose PanoGS, an effective 3DGS-based approach for 3D open vocabulary panoptic scene understanding which addresses the challenge of accurate 3D language feature learning and consistent instance-level open vocabulary segmentation. For the semantic information, we regress the 3D language features from a latent continuous parametric feature space learned by the latent pyramid tri-planes and 3D feature decoder. For the panoptic information, we adopt the language-guided graph cuts and progressive clustering strategy to construct geometrically and semantically consistent super-primitives and obtain the 3D panoptic information. Extensive experiments on commonly used datasets demonstrate that PanoGS outperforms existing state-of-the-art methods for 3D open vocabulary panoptic scene understanding.

Acknowledgment: This work was partially supported by NSF of China (No. 62425209).

References

[1] Yash Bhalgat, Iro Laina, João F. Henriques, Andrew Zisserman, and Andrea Vedaldi. N2F2: hierarchical scene understanding with nested neural feature fields. In *European Conference on Computer Vision*, pages 197–214, 2024. 1, 2

[2] Mathilde Caron, Hugo Touvron, Ishan Misra, Hervé Jégou, Julien Mairal, Piotr Bojanowski, and Armand Joulin. Emerging properties in self-supervised vision transformers. In *IEEE/CVF International Conference on Computer Vision*, pages 9650–9660, 2021. 1, 2

[3] Eric R Chan, Connor Z Lin, Matthew A Chan, Koki Nagano, Boxiao Pan, Shalini De Mello, Orazio Gallo, Leonidas J Guibas, Jonathan Tremblay, Sameh Khamis, et al. Efficient geometry-aware 3d generative adversarial networks. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 16123–16133, 2022. 2

[4] Danpeng Chen, Hai Li, Weicai Ye, Yifan Wang, Weijian Xie, Shangjin Zhai, Nan Wang, Haomin Liu, Hujun Bao, and Guofeng Zhang. PGSR: Planar-based gaussian splatting for efficient and high-fidelity surface reconstruction. *IEEE Transactions on Visualization and Computer Graphics*, pages 1–12, 2024. 1, 2

[5] Bowen Cheng, Maxwell D Collins, Yukun Zhu, Ting Liu, Thomas S Huang, Hartwig Adam, and Liang-Chieh Chen. Panoptic-DeepLab: A simple, strong, and fast baseline for bottom-up panoptic segmentation. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 12475–12485, 2020. 2

[6] Bowen Cheng, Ishan Misra, Alexander G. Schwing, Alexander Kirillov, and Rohit Girdhar. Masked-attention mask transformer for universal image segmentation. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2022. 2

[7] Manuel Dahnert, Ji Hou, Matthias Nießner, and Angela Dai. Panoptic 3d scene reconstruction from a single rgb image. *Advances in Neural Information Processing Systems*, 34: 8282–8293, 2021. 5

[8] Angela Dai, Angel X. Chang, Manolis Savva, Maciej Halber, Thomas Funkhouser, and Matthias Nießner. ScanNet: Richly-annotated 3D reconstructions of indoor scenes. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 2432–2443, 2017. 5, 6, 7

[9] Pedro F Felzenszwalb and Daniel P Huttenlocher. Efficient graph-based image segmentation. *International journal of computer vision*, 59:167–181, 2004. 4

[10] Xiao Fu, Shangzhan Zhang, Tianrun Chen, Yichong Lu, Lanyun Zhu, Xiaowei Zhou, Andreas Geiger, and Yiyi Liao. Panoptic NeRF: 3D-to-2D label transfer for panoptic urban scene segmentation. In *International Conference on 3D Vision*, pages 1–11, 2022. 2

[11] Stefano Gasperini, Mohammad-Ali Nikouei Mahani, Alvaro Marcos-Ramiro, Nassir Navab, and Federico Tombari. Panoster: End-to-end panoptic segmentation of lidar point clouds. *IEEE Robotics and Automation Letters*, 6(2):3216–3223, 2021. 2

[12] 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*, 2024. 4

[13] Jun Guo, Xiaojian Ma, Yue Fan, Huaping Liu, and Qing Li. Semantic gaussians: Open-vocabulary scene understanding with 3d gaussian splatting, 2024. 1

[14] Binbin Huang, Zehao Yu, Anpei Chen, Andreas Geiger, and Shenghua Gao. 2D gaussian splatting for geometrically accurate radiance fields. In *SIGGRAPH*, 2024. 1, 2

[15] Chenguang Huang, Oier Mees, Andy Zeng, and Wolfram Burgard. Visual language maps for robot navigation. In *IEEE/CVF International Conference on Robotics and Automation*, 2023. 1

[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), 2023. 1, 2, 3

[17] Justin Kerr, Chung Min Kim, Ken Goldberg, Angjoo Kanazawa, and Matthew Tancik. LERF: language embedded radiance fields. In *IEEE/CVF International Conference on Computer Vision*, pages 19672–19682, 2023. 2

[18] Alexander Kirillov, Kaiming He, Ross Girshick, Carsten Rother, and Piotr Dollár. Panoptic segmentation. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 9404–9413, 2019. 2, 5

[19] 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 *IEEE/CVF International Conference on Computer Vision*, pages 4015–4026, 2023. 2, 3, 4, 5, 6

[20] Abhijit Kundu, Kyle Genova, Xiaoqi Yin, Alireza Fathi, Caroline Pantofaru, Leonidas J Guibas, Andrea Tagliasacchi, Frank Dellaert, and Thomas Funkhouser. Panoptic neural fields: A semantic object-aware neural scene representation. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 12871–12881, 2022. 2

[21] Boyi Li, Kilian Q Weinberger, Serge Belongie, Vladlen Koltun, and Rene Ranftl. Language-driven semantic segmentation. In *International Conference on Learning Representations*, 2022. 1, 2, 4, 6

[22] Hai Li, Xingrui Yang, Hongjia Zhai, Yuqian Liu, Hujun Bao, and Guofeng Zhang. Vox-Surf: Voxel-based implicit surface representation. *IEEE Transactions on Visualization and Computer Graphics*, 30(3):1743–1755, 2022. 2

[23] Hai Li, Hongjia Zhai, Xingrui Yang, Zhirong Wu, Yihao Zheng, Haofan Wang, Jianchao Wu, Hujun Bao, and Guofeng Zhang. ImTooth: Neural implicit tooth for dental augmented reality. *IEEE Trans. Vis. Comput. Graph.*, 29 (5):2837–2846, 2023. 2

[24] Qi Lu, Jason Kuen, Shen Tiancheng, Gu Jiuxiang, Guo Weidong, Jia Jiaya, Lin Zhe, and Yang Ming-Hsuan. High-quality entity segmentation. In *IEEE/CVF International Conference on Computer Vision*, 2023. 4

[25] Hidenobu Matsuki, Riku Murai, Paul HJ Kelly, and Andrew J Davison. Gaussian splatting slam. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 18039–18048, 2024. 2, 3

[26] Nelson Max. Optical models for direct volume rendering. *IEEE Transactions on Visualization and Computer Graphics*, 1(2):99–108, 1995. 3

[27] 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 *European Conference on Computer Vision*, 2020. 1, 2

[28] Yuhang Ming, Xingrui Yang, Weihan Wang, Zheng Chen, Jinglun Feng, Yifan Xing, and Guofeng Zhang. Benchmarking neural radiance fields for autonomous robots: An overview. *Engineering Applications of Artificial Intelligence*, 2025. 1

[29] Gaku Narita, Takashi Seno, Tomoya Ishikawa, and Yohsuke Kaji. Panopticfusion: Online volumetric semantic mapping at the level of stuff and things. In *IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 4205–4212, 2019. 2

[30] Songyou Peng, Kyle Genova, Chiyu "Max" Jiang, Andrea Tagliasacchi, Marc Pollefeys, and Thomas Funkhouser. Openscene: 3d scene understanding with open vocabularies. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2023. 1, 2, 5, 6, 7

[31] Lorenzo Porzi, Samuel Rota Bulo, Aleksander Colovic, and Peter Kontschieder. Seamless scene segmentation. In *IEEE/CVF conference on computer vision and pattern recognition*, pages 8277–8286, 2019. 2

[32] Minghan Qin, Wanhua Li, Jiawei Zhou, Haoqian Wang, and Hanspeter Pfister. Langsplat: 3d language gaussian splatting. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 20051–20060, 2024. 1, 2, 3, 6, 7, 8

[33] Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, et al. Learning transferable visual models from natural language supervision. In *International Conference on Machine Learning*, pages 8748–8763, 2021. 1, 2, 6

[34] Jin-Chuan Shi, Miao Wang, Hao-Bin Duan, and Shao-Hua Guan. Language embedded 3d gaussians for open-vocabulary scene understanding. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 5333–5343, 2024. 1, 2, 3, 7

[35] Yawar Siddiqui, Lorenzo Porzi, Samuel Rota Bulò, Norman Müller, Matthias Nießner, Angela Dai, and Peter Kontschieder. Panoptic lifting for 3d scene understanding with neural fields. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 9043–9052, 2023. 2, 4

[36] Colton Stearns, Adam Harley, Mikaela Uy, Florian Dubost, Federico Tombari, Gordon Wetzstein, and Leonidas Guibas. Dynamic gaussian marbles for novel view synthesis of casual monocular videos. *arXiv preprint arXiv:2406.18717*, 2024. 2

[37] Julian Straub, Thomas Whelan, Lingni Ma, Yufan Chen, Erik Wijmans, Simon Green, Jakob J. Engel, Raul Mur-Artal, Carl Ren, Shobhit Verma, Anton Clarkson, Mingfei Yan, Brian Budge, Yajie Yan, Xiaqing Pan, June Yon, Yuyang Zou, Kimberly Leon, Nigel Carter, Jesus Briales, Tyler Gillingham, Elias Mueggler, Luis Pesqueira, Manolis Savva, Dhruv Batra, Hauke M. Strasdat, Renzo De Nardi, Michael Goesele, Steven Lovegrove, and Richard Newcombe. The Replica dataset: A digital replica of indoor spaces. *arXiv preprint arXiv:1906.05797*, 2019. 5, 6, 7, 8

[38] Ayça Takmaz, Elisabetta Fedele, Robert W. Sumner, Marc Pollefeys, Federico Tombari, and Francis Engelmann. OpenMask3D: Open-Vocabulary 3D Instance Segmentation. In *Advances in Neural Information Processing Systems*, 2023. 4

[39] Jiaxiang Tang, Jiawei Ren, Hang Zhou, Ziwei Liu, and Gang Zeng. Dreamgaussian: Generative gaussian splatting for efficient 3d content creation. *arXiv preprint arXiv:2309.16653*, 2023. 1, 2

[40] Vadim Tschernezki, Iro Laina, Diane Larlus, and Andrea Vedaldi. Neural feature fusion fields: 3d distillation of self-supervised 2d image representations. In *International Conference on 3D Vision*, pages 443–453, 2022. 2

[41] Matias Turkulainen, Xuqian Ren, Iaroslav Melekhov, Otto Seiskari, Esa Rahtu, and Juho Kannala. Dn-splatter: Depth and normal priors for gaussian splatting and meshing. *arXiv preprint arXiv:2403.17822*, 2024. 2

[42] Thang Vu, Kookhoi Kim, Tung M Luu, Thanh Nguyen, and Chang D Yoo. Softgroup for 3d instance segmentation on point clouds. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 2708–2717, 2022. 6, 7

[43] Qianqian Wang, Vickie Ye, Hang Gao, Jake Austin, Zhengqi Li, and Angjoo Kanazawa. Shape of motion: 4d reconstruction from a single video. *arXiv preprint arXiv:2407.13764*, 2024. 2

[44] Yu Wang, Xiaobao Wei, Ming Lu, and Guoliang Kang. Plgs: Robust panoptic lifting with 3d gaussian splatting. *arXiv preprint arXiv:2410.17505*, 2024. 2

[45] Yanmin Wu, Jiarui Meng, Haijie Li, Chenming Wu, Yahao Shi, Xinhua Cheng, Chen Zhao, Haocheng Feng, Errui Ding, Jingdong Wang, and Jian Zhang. Opengaussian: Towards point-level 3d gaussian-based open vocabulary understanding. In *Advances in Neural Information Processing Systems*, 2024. 1, 2, 3, 4, 5, 6, 7

[46] Haodong Xiang, Xinghui Li, Xiansong Lai, Wanting Zhang, Zhichao Liao, Kai Cheng, and Xueping Liu. Gaussianroom: Improving 3d gaussian splatting with sdf guidance and monocular cues for indoor scene reconstruction. *arXiv preprint arXiv:2405.19671*, 2024. 2

[47] Chi Yan, Delin Qu, Dan Xu, Bin Zhao, Zhigang Wang, Dong Wang, and Xuelong Li. GS-SLAM: Dense visual slam with 3d gaussian splatting. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 19595–19604, 2024. 2

[48] Mi Yan, Jiazhao Zhang, Yan Zhu, and He Wang. Maskclustering: View consensus based mask graph clustering for open-vocabulary 3d instance segmentation. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 28274–28284, 2024. 4

[49] Xingrui Yang, Hai Li, Hongjia Zhai, Yuhang Ming, Yuqian Liu, and Guofeng Zhang. Vox-Fusion: Dense tracking and

mapping with voxel-based neural implicit representation. In *IEEE International Symposium on Mixed and Augmented Reality*, pages 499–507, 2022. 1

[50] Ziyi Yang, Xinyu Gao, Wen Zhou, Shaohui Jiao, Yuqing Zhang, and Xiaogang Jin. Deformable 3D gaussians for high-fidelity monocular dynamic scene reconstruction. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 20331–20341, 2024. 2

[51] Mingqiao Ye, Martin Danelljan, Fisher Yu, and Lei Ke. Gaussian grouping: Segment and edit anything in 3d scenes. In *European Conference on Computer Vision*, 2024. 1, 4

[52] Taoran Yi, Jiemin Fang, Junjie Wang, Guanjun Wu, Lingxi Xie, Xiaopeng Zhang, Wenyu Liu, Qi Tian, and Xinggang Wang. Gaussiandreamer: Fast generation from text to 3d gaussians by bridging 2d and 3d diffusion models. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2024. 1

[53] Yingda Yin, Yuzheng Liu, Yang Xiao, Daniel Cohen-Or, Jingwei Huang, and Baoquan Chen. Sai3d: Segment any instance in 3d scenes. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 3292–3302, 2024. 4

[54] Zehao Yu, Torsten Sattler, and Andreas Geiger. Gaussian opacity fields: Efficient adaptive surface reconstruction in unbounded scenes. *ACM Transactions on Graphics*, 2024. 1

[55] Hongjia Zhai, Gan Huang, Qirui Hu, Guanglin Li, Hujun Bao, and Guofeng Zhang. NIS-SLAM: Neural implicit semantic RGB-D SLAM for 3D consistent scene understanding. *IEEE Transactions on Visualization and Computer Graphics*, 30(11):7129–7139, 2024. 1

[56] Hongjia Zhai, Hai Li, Xingrui Yang, Gan Huang, Yuhang Ming, Hujun Bao, and Guofeng Zhang. Vox-fusion++: Voxel-based neural implicit dense tracking and mapping with multi-maps. *arXiv preprint arXiv:2403.12536*, 2024. 2

[57] Hongjia Zhai, Xiyu Zhang, Boming Zhao, Hai Li, Yijia He, Zhaopeng Cui, Hujun Bao, and Guofeng Zhang. SplatLoc: 3d gaussian splatting-based visual localization for augmented reality. *IEEE Transactions on Visualization and Computer Graphics*, pages 1–11, 2025. 2, 3

[58] Hongjia Zhai, boming Zhao, Hai Li, Xiaokun Pan, Yijia He, Zhaopeng Cui, Hujun Bao, and Guofeng Zhang. Neuraloc: Visual localization in neural implicit map with dual complementary features. In *IEEE International Conference on Robotics and Automation*, 2025. 1

[59] Hongyu Zhou, Jiahao Shao, Lu Xu, Dongfeng Bai, Weichao Qiu, Bingbing Liu, Yue Wang, Andreas Geiger, and Yiyi Liao. Hugs: Holistic urban 3d scene understanding via gaussian splatting. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 21336–21345, 2024. 1

[60] Shijie Zhou, Haoran Chang, Sicheng Jiang, Zhiwen Fan, Zehao Zhu, Dejia Xu, Pradyumna Chari, Suya You, Zhangyang Wang, and Achuta Kadambi. Feature 3dgs: Supercharging 3d gaussian splatting to enable distilled feature fields. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 21676–21685, 2024. 1, 2, 3

[61] Zixiang Zhou, Yang Zhang, and Hassan Foroosh. Panoptic-polarnet: Proposal-free lidar point cloud panoptic segmentation. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 13194–13203, 2021. 2

[62] Runsong Zhu, Shi Qiu, Qianyi Wu, Ka-Hei Hui, Pheng-Ann Heng, and Chi-Wing Fu. Pcf-lift: Panoptic lifting by probabilistic contrastive fusion. In *European Conference on Computer Vision*, pages 92–108. Springer, 2025. 2, 4

[63] Xingxing Zuo, Pouya Samangouei, Yunwen Zhou, Yan Di, and Mingyang Li. FMGS: Foundation model embedded 3D gaussian splatting for holistic 3D scene understanding. *International Journal of Computer Vision*, pages 1–17, 2024. 1, 2, 3