GALA: Guided Attention with Language Alignment for Open Vocabulary Gaussian Splatting

Elena Alegret[1,2,5*] Kunyi Li[1,4*] Sen Wang[1,4] Siyun Liang[1]
Michael Niemeyer[3] Stefano Gasperini[1,4,6] Nassir Navab[1,4] Federico Tombari[1,3]

[1]Technical University of Munich [2]Universitat Politècnica de Catalunya [3]Google
[4]Munich Center for Machine Learning [5]ETH Zurich [6]Visualais

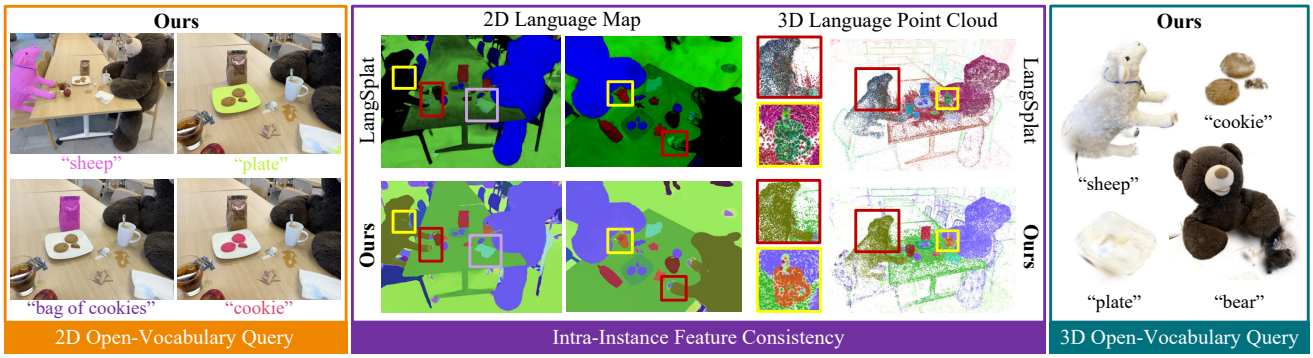

Figure 1. We present GALA, a novel 3DGS–based framework for open-vocabulary scene understanding. It delivers strong performance in both 2D and 3D open-vocabulary queries, while preserving high intra-instance feature consistency to boost segmentation quality.

Abstract

3D scene reconstruction and understanding have gained increasing popularity, yet existing methods struggle to capture fine-grained, language-aware 3D representations from 2D images. In this paper, we present GALA, a novel framework for open-vocabulary 3D scene understanding with 3D Gaussian Splatting (3DGS). GALA distills a scene-specific 3D instance feature field via self-supervised contrastive learning. To further extend this to generalized language feature fields, we introduce a core contribution of GALA, a cross-attention module with two learnable codebooks that encode view-independent semantic embeddings. This design not only ensures intra-instance feature similarity but also supports seamless 2D and 3D open-vocabulary queries. It reduces memory consumption by avoiding per-Gaussian high-dimensional feature learning. Extensive experiments on real-world datasets demonstrate GALA's remarkable open-vocabulary performance on both 2D and 3D.

*Equal contribution.

1. Introduction

Understanding 3D scenes is a central challenge in 3D computer vision (3DV), with wide-ranging applications in autonomous driving [3, 8, 40, 41], robotics [1, 11, 19, 24], and augmented or virtual reality [5, 34]. Open-vocabulary scene understanding not only enables robots to perceive and reason about the world but also opens new possibilities for intuitive human-robots interaction, allowing users to explore and query scenes through natural language. This integration of spatial understanding with language grounding represents a promising direction towards more intelligent systems.

Neural Radiance Fields (NeRF) [2, 25, 26, 43] offer the potential to store additional semantic information within the field. Several methods [14, 15, 45] extend NeRF by distilling semantic or language features from 2D images. However, NeRF-based methods suffer from inefficient encoding and incur high computational costs for training and rendering. 3D Gaussian Splatting (3DGS) [9, 13, 20, 23, 27, 46] provides an explicit and more efficient alternative by representing scenes with a set of 3D Gaussian shape primitives. Subsequent works [12, 21, 31, 32, 47] incorporate feature

attributes into these Gaussians, enabling semantic feature rasterization and language-based interactions.

Robotic systems with limited computing resources such as those for navigation, are often performed in 2D, even though the robots operate in a 3D world. Therefore, 2D perception remains essential. Recent works [31, 47] address this by distilling high-dimensional 2D language features [16, 33] into 3D Gaussians through compressing high-dimensional language features into low-dimensional representations, followed by novel view synthesis to enable arbitrary-view 2D open-vocabulary querying. However, such compression inevitably leads to information loss, and their segmentation results exhibit low intra-instance consistency and blurred object boundaries, which hinder accurate semantic segmentation.

Instead of prioritizing efficient 2D open-vocabulary segmentation, some works focus on enhancing 3D scene semantics. However, storing high-dimensional language features for each Gaussian is computationally time- and memory-intensive. Recent methods [18, 42] address this issue with clustering: Gaussians are grouped into clusters, each assigned a low-dimensional scene-specific feature, and are matched to preprocessed per-instance language features via 2D–3D associations. Yet, purely KNN-based clustering without explicit supervision can cause one cluster to span multiple instances or split a single instance, leading to misalignment and degraded segmentation performance. Others [6, 12] average multi-view language features without training, achieving strong 3D reconstruction but offering limited or memory-heavy 2D semantic rendering, making them unsuitable for real-time robotics and navigation.

Although reconstructing and perceiving the world in 3D is important, interacting with it in 2D is often the most efficient strategy for robotics [1, 10, 17], existing approaches tend to focus only on one side of the problem. We propose a Guided Attention method with Language Alignment Gaussian Splatting (GALA), a novel framework that enables both 2D and 3D open-vocabulary scene understanding demonstrating its broad applicability to diverse perception tasks, as illustrated in Figure 1. The key idea is to enforce instance-consistent semantics: instead of storing noisy or redundant per-Gaussian language features, GALA learns to associate each Gaussian with a shared instance-level language embedding, ensuring that the semantics of each instance remain consistent not only across different spatial locations and viewpoints but also within the 3D scene. Our main contributions can be summarized as follows:

- We propose to store per-instance semantics via codebooks, associating each instance with a language embedding and ultimately generating intra-instance consistent semantic features for better segmentation.
- By employing an attention mechanism that maps each Gaussian feature to its corresponding instance, we enable effective 2D and 3D open-vocabulary segmentation.
- We improve the segmentation with an attention-weighted entropy loss, which encourages a clear one-to-one mapping between Gaussian instance features and codebook embeddings.

Extensive experiments on public real-world datasets, LERF-OVS [14] and ScanNet-v2 [7], demonstrate the effectiveness of GALA on both 2D and 3D semantic segmentation and open-vocabulary localization compared to the state-of-the-art. The code and models will be released upon acceptance.

2. Related Works

2.1. Zero-Shot 2D Scene Understanding

The success of 2D visual foundation models has been demonstrated across a wide range of vision tasks, which enhances both perceptual and reasoning abilities. CLIP [33] aligns image and text features through contrastive learning, enabling robust cross-modal understanding in a shared embedding space. DINO [28], a self-supervised Vision Transformer, learns rich semantic representations from unlabeled images, capturing object boundaries and scene layouts. Building on these models, Grounding DINO [22] extends DINO with open-vocabulary detection capabilities guided by textual queries, through tight visual-language fusion. SAM [16], a promptable segmentation model, enables zero-shot instance segmentation with impressive generalization. Grounded SAM [35] combines SAM with Grounding DINO to support arbitrary text-driven semantic segmentation and detection. APE [36] introduces a unified visual perception framework for tasks like segmentation and grounding, using lightweight visual-language fusion for efficient and generalizable performance. However, these powerful models are inherently limited to 2D image understanding, restricting their applicability in tasks requiring holistic 3D scene understanding.

2.2. Open-Vocabulary 3D Scene Understanding

Understanding 3D scenes requires consistent semantic reasoning across multiple views and spatial dimensions. Recent efforts have explored transferring powerful language features from 2D models into 3D representations to allow robots to perceive the world like humans. OpenScene [29] distills CLIP features into 3D point clouds for zero-shot segmentation and language queries, but suffers from limited spatial resolution and reduced generalization due to point-based representation. More recent methods [4, 14, 15, 45] integrate semantics into continuous neural radiance field by distilling 2D language features, enabling open-vocabulary 3D understanding. However, NeRFs remain slow to render, depend heavily on high-quality 2D masks, and struggle with scalability due to volumetric computation.

In contrast, 3D Gaussian Splatting (3DGS) provides an explicit and efficient representation better suited for real-time 3D understanding. LangSplat [31] applies hierarchical feature distillation by assigning each Gaussian a low-dimensional feature that is rasterized into a 2D feature map. A pretrained autoencoder is used to compress high-dimensional language features for supervision. Similarly, Feature3DGS [47] leverages a convolutional neural network (CNN) for feature dimension lifting. While both methods reduce the dimensionality of the supervision signal, this compression inevitably causes information loss. Furthermore, they learn per-Gaussian semantic features without enforcing intra-instance feature consistency, which may lead to ambiguous object representations and hinder robotic interaction and navigation. OpenGaussian [42] and InstanceGaussian [18] place greater emphasis on 3D awareness by enabling point-level 3D segmentation through hierarchical feature clustering and 3D–2D feature association, mapping scene-specific instance features to language features. However, misalignment in this mapping can cause significant performance drops.

Rather than training a semantic feature field per scene, Dr. Splat [12] and Occam's LGS [6] propose an aggregation method that averages multi-view language features in a single forward pass, greatly improving efficiency. Although these methods improve 3D semantic reconstruction, generating accurate 2D semantic maps remains crucial for robotics, enabling fast and reliable perception from onboard camera images. SuperGSeg [21] clusters thousands of Gaussians into SuperGaussians sharing language embeddings, enabling efficient high-dimensional feature rendering and improving performance. However, its MLP-based cluster update is complex and may lack semantic coherence, sometimes grouping irrelevant or noisy points. Moreover, the K-Nearest Neighbors (KNN)-based initialization depends on point density, so sparse regions can cause a SuperGaussian to span multiple objects with conflicting semantics, degrading segmentation quality. GOI [32] and CCL-LGS [38] both introduce a single trainable feature codebook to store language embeddings and use a multi-layer perceptron (MLP) to predict discrete codebook indices for the rasterized 2D feature maps. While this approach compresses semantics spatially rather than dimensionally preserving semantic richness, the MLP applies fixed weights uniformly across all input elements, lacking the flexibility to dynamically prioritize important information. This limitation makes it less effective at capturing context-dependent relevance compared to attention mechanisms.

Therefore, we propose a dual-codebook design combined with a guided cross-attention module. Our method computes similarity scores for soft, continuous assignments between Gaussian features and codebook embeddings, enabling instance-level semantics in a differentiable manner. Despite relying on 2D supervision, the fully linear attention and rasterization modules enhance generalization from 2D tasks to 3D tasks and effectively reduce the multi-view inconsistencies found in prior work.

3. Preliminaries

3D Gaussian Splatting (3DGS) [13] employs a set of 3D points to effectively render images from given viewpoints, each characterized by a Gaussian function with 3D mean $\mu_i \in \mathbb{R}^3$, covariance matrix $\Sigma_i \in \mathbb{R}^{3\times3}$, opacity value $\alpha_i \in \mathbb{R}$, RGB color value $\mathbf{c}_i \in \mathbb{R}^3$, and sometimes with feature value $\mathbf{m}_i \in \mathbb{R}^d$:

$$\sigma_i(\mathbf{x}) = \alpha_i * \exp\left(-\frac{1}{2}(\mathbf{x}-\mu_i)^T\Sigma_i^{-1}(\mathbf{x}-\mu_i)\right). \quad (1)$$

Given a 3D position $\mathbf{x}$, $\sigma_i(\mathbf{x})$ represents current opacity value contributed by the i-th Gaussian. To facilitate optimization, $\Sigma_i = R_iS_iS_i^TR_i^T$ is factorized into the product of a scaling matrix S_i, represented by scale factors $\mathbf{s}_i \in \mathbb{R}^3$, and a rotation matrix R_i encoded by a quaternion $\mathbf{r}_i \in \mathbb{R}^4$. Color value $\hat{\mathbf{C}}(\mathbf{u})$ and feature value $\hat{\mathbf{M}}(\mathbf{u})$ at pixel $\mathbf{u}$ are rendered by N projected and ordered Gaussians using point-based α-blending:

$$\{\hat{\mathbf{C}}, \hat{\mathbf{M}}\}(\mathbf{u}) = \sum_{i\in N} T_i\sigma_i \times \{\mathbf{c}_i, \mathbf{m}_i\}, \quad (2)$$

where $T_i = \prod_{j=1}^{i-1}(1-\sigma_j)$. Scaffold-GS [23] introduces a neural variant of 3DGS by voxelizing a set of point clouds as anchor points $V \in \mathbb{R}^{N\times3}$. Each anchor point $\mathbf{v}_i \in V$ is associated with a feature $\mathbf{f}_i \in \mathbb{R}^d$, scaling factor $l_i \in \mathbb{R}^3$ and K learnable offsets $\{\mathcal{O}_{i,k} \in \mathbb{R}^3 \mid k = 0, \ldots, K-1\}$. Then K neural Gaussians $\{\mu_{i,0}, \ldots, \mu_{i,K-1}\} = \mathbf{v}_i + \{\mathcal{O}_{i,0}, \ldots, \mathcal{O}_{i,K-1}\} \cdot l_i$ are generated from a given anchor point $\mathbf{x}_v$. The remaining attributes of each Gaussian $\mathbf{g}_i \in \{\alpha_{i,k}, \mathbf{c}_{i,k}, R_{i,k}, S_{i,k}, \mathbf{m}_{i,k}\}$ are predicted as:

$$\{\mathbf{g}_{i,0}, \ldots, \mathbf{g}_{i,K-1}\} = \mathcal{F}_{\mathbf{g}}(\mathbf{f}_i, \delta_i, \vec{\mathbf{d}}_i), \quad (3)$$

where $\delta_i = \|\mathbf{v}_i - \mathbf{x}_c\|$, $\vec{\mathbf{d}}_i = \frac{\mathbf{v}_i - \mathbf{x}_c}{\|\mathbf{v}_i - \mathbf{x}_c\|}$, $\mathbf{x}_c$ is the camera center, and $\mathcal{F}_{\mathbf{g}}$ is corresponding attribute decoder.

4. Methods

As shown in Figure 2, our method builds on neural Gaussian Splatting [23] with two-stage training: (1) self-supervised reconstruction of scene geometry and a scene-specific instance feature field, and (2) rendering these features to 2D and mapping them to generalized language features via guided attention with dual learnable codebooks. The linear attention design enables seamless segmentation in both 2D and 3D using only 2D training, while the per-instance codebooks and attention-weights entropy loss enforce one-to-one mappings, enhancing intra-instance feature consistency.

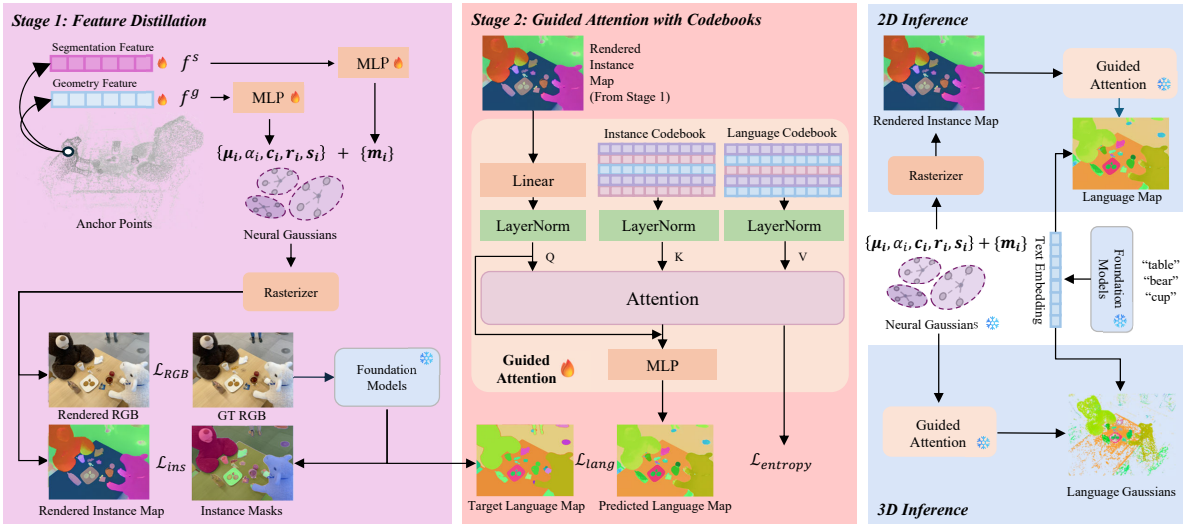

Figure 2. **Overview of GALA.** In Stage 1, we reconstruct the 3D scene and distill a scene-specific feature field in a self-supervised manner. In Stage 2, a rasterized instance feature map is used to train a Guided Attention module, which learns to map the scene-specific feature field to a generalized language field via two learnable codebooks. During inference (right), GALA supports open-vocabulary querying and segmentation in both 2D (top) and 3D (bottom).

4.1. Scene Reconstruction and Feature Distillation

SuperGSeg [21] builds on Scaffold-GS [23] to perform joint 3D reconstruction and scene-specific instance and hierarchical feature distillation, where these features are used for clustering. In contrast, our method does not rely on hierarchical features for clustering and instead focuses solely on instance learning in a self-supervised manner. Consequently, each anchor point in our method is assigned to two types of features. Using Eq. 3, a geometry feature $\mathbf{f}_i^g \in \mathbb{R}^{d_g}$ is decoded into K Gaussian attributes $\{\alpha_{i,k}, \mathbf{c}_{i,k}, R_{i,k}, S_{i,k}\}$, and a segmentation feature $\mathbf{f}_i^s \in \mathbb{R}^{d_{\text{seg}}}$ is decoded into K instance features $\mathbf{m}_{i,k} \in \mathbb{R}^{d_{\text{ins}}}$. These attributes and features are then rasterized as a rendered color map $\hat{\mathbf{C}} \in \mathbb{R}^{H\times W\times 3}$ and a instance feature map $\hat{\mathbf{M}} \in \mathbb{R}^{H\times W\times d_{\text{ins}}}$ through Eq. 2.

LangSplat [31] and Feature3DGS [47] learn per-Gaussian semantic features independently, without any instance-level constraints. Our method adopts a self-supervised method [42, 45] to distill a scene-specific instance field with instance contrastive learning. We first generate a set of instance masks $\{m_i \in \mathbb{R} \mid i = 0, \ldots, \mathcal{M}\}$ for each view using Segment Anything Model (SAM) [16]. For a given instance mask m_i, we denote each pixel feature within the mask as $\{\hat{m}_{i,j} \in \mathbb{R}^{d_{\text{ins}}} \mid j = 1, \ldots, n\}$, where subscript i denotes the mask index and subscript j denotes the pixel index. We compute the mean feature value within the mask as $\bar{m}_i \in \mathbb{R}^{d_{\text{ins}}}$. To distill the 3D instance field in a self-supervised manner and enhance intra-instance feature similarity, we employ contrastive learning to pull features within the same mask closer together, while pushing features from different masks further apart:

$$\mathcal{L}_{\text{ins}} = \frac{1}{\mathcal{M}} \sum_{i=1} \sum_{j=1} -\log \frac{\exp(\hat{m}_{i,j} \cdot \bar{m}_i / \tau_i)}{\sum_{q\neq i}^{\mathcal{M}} \exp(\hat{m}_{i,j} \cdot \bar{m}_q / \tau_q)}. \quad (4)$$

Therefore, the overall objective function for the first stage is:

$$\mathcal{L}_1 = \mathcal{L}_{\text{RGB}} + \lambda_{\text{ins}} \mathcal{L}_{\text{ins}}, \quad (5)$$

where λ_{ins} is the penalty coefficient. And $\mathcal{L}_{\text{RGB}} = 0.8 \times |\mathbf{C} - \hat{\mathbf{C}}| + 0.2 \times SSIM(\mathbf{C} - \hat{\mathbf{C}})$ is the photometric loss [13] where $\mathbf{C}$ is the ground-truth color image.

4.2. Semantic Codebooks

Prior works such as OpenGaussian [42], InstanceGaussian [18], and SuperGSeg [21] adopt a bottom-up approach: they first cluster low-level features to form clusters and then learn instance-level segmentation by aggregating these clusters. However, this strategy can lead to several issues, including over-segmentation, one single cluster representing multiple distinct objects, or different parts of the same object being assigned to separate clusters. To address this, we introduce a codebook module designed to represent each instance in the scene with a unique embedding. A codebook consists of N_c learnable embeddings, where N_c approximates the number of instances in the scene. Specifically, we define an Instance Codebook $\mathcal{C}_{ins} \in R^{N_c \times d_{ins}}$, where each entry captures a distinct instance-level representation. In parallel, we define a Language Codebook $\mathcal{C}_{lang} \in R^{N_c \times d_c}$ which stores language embeddings with a one-to-one correspondence to the entries in $\mathcal{C}_{inst}$. Each codebook entry is

intended to represent a unique instance in the scene. The proposed codebooks decouple semantics from spatial positions and allow for unambiguous, per-instance embedding assignments, ensuring intra-instance feature similarity.

4.3. Guided Attention with Codebooks

Perceiving the world through human language is a key goal of 3D scene understanding, for which a purely scene-specific feature field is insufficient. Prior works [18, 42] attempt to align low-dimensional features with high-dimensional language semantics via 2D–3D associations, while others [31, 47] compress high-dimensional supervision to reduce overhead. However, these approaches are either designed for 3D or 2D segmentation tasks, suffering from information loss, or leading to limited generalization. Our method introduces a guided cross-attention module with codebooks proposed in Section 4.2 that maps scene-specific features to the generalized language field, enabling both 2D and 3D open-vocabulary queries.

Attention with Codebooks. An attention module [39] is adopted with learnable codebooks and residual connections. We use the rasterized instance feature map $\hat{\mathbf{M}}$ as the query Q, the instance codebook $\mathcal{C}_{ins}$ as the key K and the language codebook $\mathcal{C}_{lang}$ as the value V:

$$\hat{\mathbf{A}} = \mathcal{A}(\hat{\mathbf{M}}) = Attn(Q, K, V) + Q \in \mathbb{R}^{HW \times d_c}, \quad (6)$$

$$\hat{\mathbf{L}} = \mathcal{F}_{lift}(\hat{\mathbf{A}}) \in \mathbb{R}^{HW \times d_{lang}}, \quad (7)$$

where $Q = \mathcal{N}(\hat{\mathbf{M}} \times W^Q), K = \mathcal{N}(C_{\text{ins}}), V = \mathcal{N}(C_{\text{lang}})$, $\mathcal{N}$ represents layer normalization opertor which is applied to avoid scale discrepancies, W^Q is the linear transformation which is applied to project the original input into a same space of instance codebook and $\mathcal{F}_{lift}$ is an MLP to lift the feature dimensionality. During training, we apply only 2D supervision with cosine similarity loss between the predicted language map $\hat{\mathbf{L}}$ and the preprocessed ground-truth language map $\mathbf{L}$:

$$\mathcal{L}_{\text{lang}} = 1 - \cos(\hat{\mathbf{L}}, \mathbf{L}). \quad (8)$$

It is worth noting that the attention operation $\mathcal{A}$ defined in Eq. 6 is linear. As the rasterization in Eq. 2 involves a weighted summation, applying $\mathcal{A}$ to the 2D rasterized features is mathematically equivalent to applying it directly to the underlying 3D Gaussians:

$$\mathcal{A}(\hat{\mathbf{M}}) = \mathcal{A}\big(\sum T_i \sigma_i \times \mathbf{m}_i\big) = \sum T_i \sigma_i \times \mathcal{A}(\mathbf{m}_i). \quad (9)$$

This property allows us to train the codebooks solely using 2D feature maps as supervision, and during inference, however, the same model can be directly applied to the 3D Gaussians, enabling open-vocabulary semantic queries in both 2D and 3D space. By compacting per-Gaussian's semantic features into per-instance embeddings, we not only reduce training costs but also enforce intra-instance feature consistency in both 2D and 3D.

Probability Guidance. OpenGaussian [42] adopts two-level clustering with positional embedding to model instance-level representations. However, without explicit supervision, it struggles to establish a one-to-one correspondence between instances and clusters. Our method leverages attention weights to guide a clear one-to-one mapping between instances and codebook embeddings. The attention weights:

$$P = softmax(QK^\top / \sqrt{d_{ins}}) \in \mathbb{R}^{HW \times N_c}, \quad (10)$$

indicate the relevance probability of each codebook embedding with respect to each feature query. To encourage a one-to-one correspondence, we apply the entropy loss on the attention weights:

$$\mathcal{L}_{\text{entropy}} = -\sum_{j=1} p_j \log(p_j), \quad (11)$$

where $p_j \in P$ represents the attention probability distribution over the N_c codebook entries for feature query j. This enforces the probability distribution for each query to be unimodal, meaning each query is associated with a single codebook embedding, ensuring that each instance corresponds to only one embedding in the codebook.

Therefore, the overall objective function of the second stage is:

$$\mathcal{L}_2 = \mathcal{L}_{\text{lang}} + \lambda_{\text{ent}} \mathcal{L}_{\text{entropy}}, \quad (12)$$

where λ_{ent} is the penalty coefficient.

5. Experiments

5.1. Experimental Setup

Datasets. We comprehensively evaluate our method on two real-world datasets: ScanNet-v2 [7] and LERF-OVS [14]. Following OpenGaussian [42], 8 scenes are selected from the ScanNet-v2.

Baselines. We compare our method in both 2D and 3D with LERF [14], LangSplat [31], Feature-3DGS [47], GS-Grouping [44], LEGaussians [37], GOI [32], SuperGSeg [21] and OpenGaussian [42].

Metrics. We follow common practice and report open-vocabulary segmentation and object selection evaluation with mean Intersection-over-Union (mIoU) for segmentation accuracy and mean accuracy (mAcc) for localization accuracy. While understanding the world in 3D is essential, perceiving it in 2D offers a more efficient pathway for real-time performance in robotics. Therefore, we report our performance both in 2D and 3D to demonstrate the broad applicability of our method to diverse perception tasks.

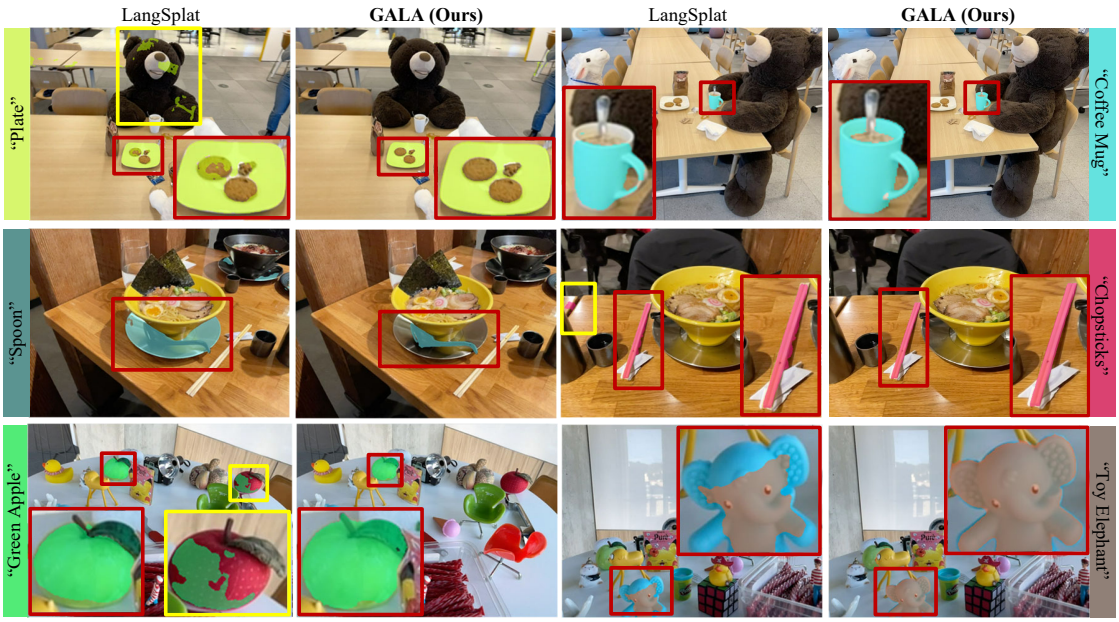

Figure 3. **Qualitative Results of 2D Open-Vocabulary Query.** We visualize 2D Open-Vocabulary query results on LERF-OVS dataset [14]. LangSplat fails to localize objects accurately, leading to mismatching or incomplete masks. Our method delivers precise and consistent queries across diverse queries.

		Mean		*Figurines*		*Teatime*		*Ramen*		*Waldo_kitchen*	
Eval.	**Method**	mIoU ↑	mAcc ↑	mIoU	mAcc	mIoU	mAcc	mIoU	mAcc	mIoU	mAcc
2D	LERF [14]	37.40	73.60	38.60	75.00	45.00	84.80	28.20	62.00	37.90	72.70
	LangSplat [31]	51.40	84.30	44.70	80.40	65.10	88.10	51.20	73.20	44.50	**95.50**
	Feature-3DGS [47]	45.70	77.00	58.80	77.20	40.50	73.40	43.70	69.80	39.60	87.60
	GS-Grouping [44]	46.30	76.50	60.90	75.00	40.00	74.30	45.50	68.60	38.70	88.20
	LEGaussians [37]	46.90	77.20	60.30	75.60	40.80	75.20	46.00	67.50	39.40	90.30
	GAGS [30]	54.12	81.66	53.59	78.57	60.29	88.14	46.81	69.01	55.80	90.91
	GOI [32]	50.60	**84.40**	**63.70**	**88.60**	44.50	82.90	**52.60**	**75.50**	41.40	90.40
	Ours	**55.49**	73.43	59.35	82.14	**76.73**	**88.14**	35.13	50.70	50.75	72.73
3D	LangSplat [31]	9.66	12.41	10.16	8.93	11.38	20.34	7.92	11.27	9.18	9.09
	LEGaussians [37]	16.21	23.82	17.99	23.22	19.27	27.12	15.79	26.76	11.78	18.18
	OpenGaussian [42]	**38.36**	51.43	39.29	55.36	**60.44**	76.27	**31.01**	**42.25**	22.70	31.82
	SuperGSeg [21]	35.94	52.02	43.68	60.71	55.31	77.97	18.07	23.94	26.71	45.45
	Ours	36.71	**59.71**	**45.25**	**69.64**	53.27	**84.75**	17.08	25.35	**31.22**	**59.09**

Table 1. **2D and 3D Evaluation on LERF-OVS.** We report mIoU and mAcc on the LERF-OVS dataset [14]. Note that OpenGaussian [42] and SuperGSeg [21] by default do not report 2D evaluation. LERF [14], Feature-3DGS [47], GS-Grouping [44], GAGS [30] and GOI [32] by default do not support 3D evaluation.

Implementation Details. We perform single-GPU training (NVIDIA RTX 3090). For stage 1, we train 30,000 iterations with $\lambda_{\text{ins}} = 0.001$ and for stage 2 we train 15,000 iterations with $\lambda_{\text{ent}} = 10$. We set the dimension of both instance codebook and language codebook as $d_{ins} = d_c = 16$. We use SAM [16] and CLIP [33] to preprocess the ground-truth language map, and set $d_{lang} = 512$. For more implementation details, please refer to the supp. mat..

5.2. 2D Evaluation

Table 1 presents the 2D results on the LERF-OVS dataset. We report both per-scene and average evaluations, where our method achieves the highest average mIoU (55.49%) among all existing approaches. In scenes with clearly separated objects, such as *Teatime*, our method delivers precise performance in both open-vocabulary segmentation and localization, achieving 76.73% mIoU and 88.14% mAcc. Our method also performs robustly in more cluttered environments like *Waldo_Kitchen*, attaining 50.75% mIoU, where

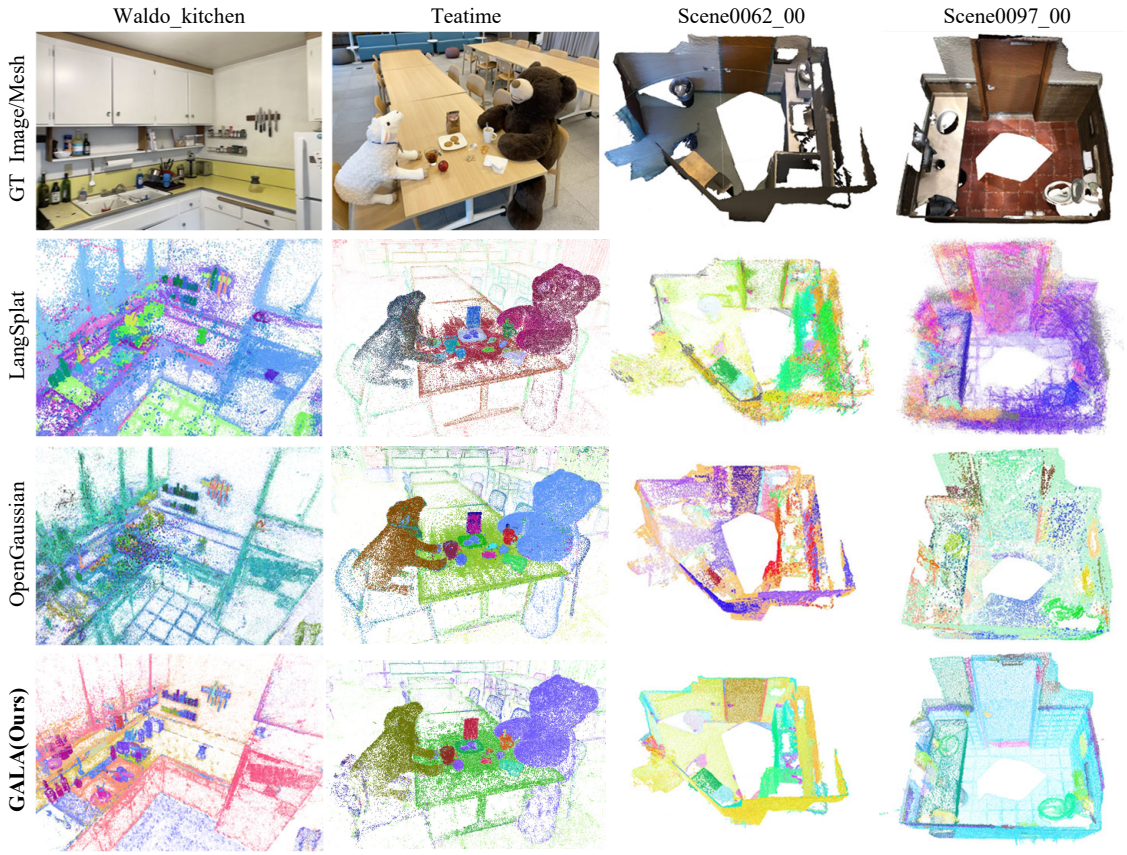

Figure 4. **Qualitative Results of 3D Open-Vocabulary Segmentation.** We visualize the language feature point cloud on LERF-OVS [14] and ScanNet-v2 dataset [7] by compressing the features into the RGB point cloud. Note that the colors for visualization are consistent only within each method and not method-to-method.

it better distinguishes complex domestic objects compared to LangSplat and GOI. Figure 3 demonstrates that our method can accurately distinguish the *"Green Apple"* without ambiguity, whereas LangSplat incorrectly selects both the green and red apples. Overall, our method achieves precise object segmentation with sharp and well-defined boundaries.

5.3. 3D Evaluation

3D Evaluation on LERF-OVS. Following the evaluation protocol of LangSplat [31], Table 1 showcases the strong performance of our method in 3D segmentation and localization on the LERF-OVS dataset. Thanks to the linearity of the proposed attention module, our method, trained exclusively with 2D supervision, generalizes seamlessly to 3D tasks without any architectural modifications. Our method even outperforms the 3D-only method OpenGaussian on *Figurines* and *Waldo kitchen*, which, however, cannot easily produce 2D segmentation outputs. In contrast, the 2D-only method LangSplat struggles with 3D evaluation, as it is trained solely with 2D supervision and lacks 3D-aware segmentation. We also visualize the feature point cloud in Figure 4. Our method achieves both better geometry reconstruction and 3D segmentation compared with LangSplat [31] and OpenGaussian [42].

3D Evaluation on ScanNet-v2. Table 2 reports the 3D point cloud segmentation results on the ScanNet-v2 dataset, as ScanNet-v2 provides ground-truth semantic point cloud. We present the mean mIoU and mAcc across eight selected scenes containing different numbers of classes. Our method consistently outperforms OpenGaussian on all metrics, delivering strong 3D reconstruction and segmentation accuracy alongside high-quality 3D localization. Figure 4 visualizes the language-featured point clouds. By default, OpenGaussian does not densify Gaussians on ScanNet-v2, resulting in sparse features and lower appearance quality. Our method surpasses OpenGaussian both quantitatively and qualitatively.

5.4. Intra-Instance Feature Consistency

Previous methods learn semantic features per Gaussian or cluster, causing variations across positions and view-

Method	19 Classes		15 Classes		10 Classes	
	mIoU ↑	mAcc ↑	mIoU ↑	mAcc ↑	mIoU ↑	mAcc ↑
LangSplat [31]	2.94	11.63	3.80	13.98	6.60	22.24
OpenGaussian [42]	15.47	26.04	17.42	28.82	23.46	37.73
Ours	**21.54**	**37.47**	**25.20**	**42.06**	**35.85**	**57.02**

Table 2. **3D Evaluation on ScanNet-v2.** We report the average 3D mIoU and mAcc on 8 scenes of the ScanNet-v2 dataset [7].

Figure 5. **Intra-Instance Feature Consistency.** We visualize the rendered language feature map and show that our method provides a consistent intra-instance feature map and a clear boundary, which enhances the segmentation performance.

points. Our method addresses this issue in Stage 1 through instance-level contrastive learning, and further reinforces feature consistency via a per-instance codebook design. As shown in Figure 4, on the *Teatime* and *Waldo_kitchen* scenes, the feature point clouds produced by LangSplat are highly noisy. On *Scene0097_00*, the door features from LangSplat are difficult to distinguish, and OpenGaussian oversegments the floor. Figure 5 visualizes the results with rendered 2D language feature maps. Our method yields homogeneous feature maps with well-defined boundaries.

5.5. Ablation Study

All ablations are on *Teatime* of LERF-OVS [14].

Ablation on Codebook. Table 3 shows an ablation on the number of codes. *Teatime* contains around 64 instances; therefore, the best performance is achieved with 64 codes, matching the number of instances and enabling a near one-to-one mapping between embeddings and objects. Figure 6 shows that too few codes cause semantic ambiguity. The codebook size matching the expected number of instances achieves the best balance of accuracy and efficiency.

Ablation on Attention Module. We also report ablation on the structure of the proposed attention module. In Table 4, we show results that a) with lifting MLP Eq. 7 only, b) with attention Eq. 6 only and set the language codebook as $\mathcal{C}_{lang} \in R^{N_c \times 512}$, c) the attention together with lifting MLP but without residual connection $\mathcal{F}_{lift}(\mathcal{A}(Q,K,V))$, e) our full model. We find that our full model achieves the best overall performance. In case b), applying the attention module directly without the MLP leads to high computational cost and convergence difficulties. Comparing cases c) and d), we observe that introducing a residual connection significantly improves training stability.

Figure 6. **Ablation on the Number of Codes.** We visualize an embedding from the codebook as the semantic mask. With codes number $N_c = 16$, code 14 represents both the sheep and plate in the *teatime* scene of LERF-OVS. With $N_c = 64$, our method clearly isolates the sheep, demonstrating improved instance separation.

$N_c =$	16	32	64	128
mIoU ↑	60.22	71.65	**76.73**	68.33
mAcc ↑	72.88	83.05	**88.14**	83.05

Table 3. **Ablation on Number of Codes.**

#	MLP	Attn	Res.	Prob.	mIoU ↑	mAcc ↑
a)	✓				72.60	86.44
b)		✓			35.48	42.37
c)	✓	✓			74.25	88.13
d)	✓	✓	✓		75.02	86.44
e)	✓	✓	✓	✓	**76.73**	**88.14**

Table 4. **Ablation on Attention and Probability Guidance.**

Ablation on Probability Guidance. In Table 4, we show results that d) without probability guidance. Our guided attention model is better able to assign distinct embeddings to separate object instances, leading to more accurate and interpretable segmentation. The right column of Figure 6 visualizes a selected embedding from our proposed codebook, showing that each embedding indeed captures meaningful instance-level semantics. For more ablations and runtime analysis, please refer to **supp. mat.**.

6. Conclusions

We presented GALA, a framework for open-vocabulary 3D scene understanding using 3D Gaussian Splatting. By combining self-supervised instance-level feature distillation with a cross-attention module and learnable codebooks, GALA produces consistent, view-independent semantic embeddings, supports 2D and 3D open-vocabulary queries, and reduces memory usage. Experiments on real-world datasets demonstrate its effectiveness in generating reliable and efficient 3D and 2D feature representations.

References

[1] Peter Anderson, Qi Wu, Damien Teney, Jake Bruce, Mark Johnson, Niko Sünderhauf, Ian Reid, Stephen Gould, and Anton Van Den Hengel. Vision-and-language navigation: Interpreting visually-grounded navigation instructions in real environments. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 3674–3683, 2018. 1, 2

[2] Jonathan T Barron, Ben Mildenhall, Matthew Tancik, Peter Hedman, Ricardo Martin-Brualla, and Pratul P Srinivasan. Mip-nerf: A multiscale representation for anti-aliasing neural radiance fields. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 5855–5864, 2021. 1

[3] Holger Caesar, Varun Bankiti, Alex H. Lang, et al. nuscenes: A multimodal dataset for autonomous driving. *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2020. 1

[4] Jiazhong Cen, Zanwei Zhou, Jiemin Fang, Wei Shen, Lingxi Xie, Dongsheng Jiang, Xiaopeng Zhang, Qi Tian, et al. Segment anything in 3d with nerfs. *Advances in Neural Information Processing Systems*, 36:25971–25990, 2023. 2

[5] Jiaqi Chen, Ruoxi Zhao, et al. Scenear: Learning to reconstruct 3d indoor scenes for augmented reality. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2023. 1

[6] Jiahuan Cheng, Jan-Nico Zaech, Luc Van Gool, and Danda Pani Paudel. Occam's lgs: An efficient approach for language gaussian splatting. *arXiv preprint arXiv:2412.01807*, 2024. 2, 3

[7] Angela Dai, Angel X Chang, Manolis Savva, Maciej Halber, Thomas Funkhouser, and Matthias 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. 2, 5, 7, 8, 1, 4

[8] Andreas Geiger, Philip Lenz, and Raquel Urtasun. Are we ready for autonomous driving? the kitti vision benchmark suite. *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2012. 1

[9] Binbin Huang, Zehao Yu, Anpei Chen, Andreas Geiger, and Shenghua Gao. 2d gaussian splatting for geometrically accurate radiance fields. In *ACM SIGGRAPH 2024 conference papers*, pages 1–11, 2024. 1

[10] Chenguang Huang, Oier Mees, Andy Zeng, and Wolfram Burgard. Visual language maps for robot navigation. *arXiv preprint arXiv:2210.05714*, 2022. 2

[11] Nathan Hughes, Yun Chang, Siyi Hu, Rajat Talak, Rumaia Abdulhai, Jared Strader, and Luca Carlone. Foundations of spatial perception for robotics: Hierarchical representations and real-time systems. *The International Journal of Robotics Research*, 43(10):1457–1505, 2024. 1

[12] Kim Jun-Seong, GeonU Kim, Kim Yu-Ji, Yu-Chiang Frank Wang, Jaesung Choe, and Tae-Hyun Oh. Dr. splat: Directly referring 3d gaussian splatting via direct language embedding registration. In *Proceedings of the Computer Vision and Pattern Recognition Conference*, pages 14137–14146, 2025. 1, 2, 3, 4

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

[14] Justin Kerr, Chung Min Kim, Ken Goldberg, Angjoo Kanazawa, and Matthew Tancik. Lerf: Language embedded radiance fields. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 19729–19739, 2023. 1, 2, 5, 6, 7, 8

[15] Chung Min Kim, Mingxuan Wu, Justin Kerr, Ken Goldberg, Matthew Tancik, and Angjoo Kanazawa. Garfield: Group anything with radiance fields. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 21530–21539, 2024. 1, 2

[16] Alexander Kirillov, Eric Mintun, Nikhila Ravi, Hanzi Mao, Chloe Rolland, Laura Gustafson, Tete Xiao, Spencer Whitehead, Alexander C Berg, Wan-Yen Lo, et al. Segment anything. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 4015–4026, 2023. 2, 4, 6, 1

[17] Xiaohan Lei, Min Wang, Wengang Zhou, and Houqiang Li. Gaussnav: Gaussian splatting for visual navigation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2025. 2

[18] Haijie Li, Yanmin Wu, Jiarui Meng, Qiankun Gao, Zhiyao Zhang, Ronggang Wang, and Jian Zhang. Instancegaussian: Appearance-semantic joint gaussian representation for 3d instance-level perception. In *Proceedings of the Computer Vision and Pattern Recognition Conference*, pages 14078–14088, 2025. 2, 3, 4, 5

[19] Kunyi Li, Michael Niemeyer, Nassir Navab, and Federico Tombari. Dns-slam: Dense neural semantic-informed slam. In *2024 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 7839–7846. IEEE, 2024. 1

[20] Yanyan Li, Chenyu Lyu, Yan Di, Guangyao Zhai, Gim Hee Lee, and Federico Tombari. Geogaussian: Geometry-aware gaussian splatting for scene rendering. In *European conference on computer vision*, pages 441–457. Springer, 2024. 1

[21] Siyun Liang, Sen Wang, Kunyi Li, Michael Niemeyer, Stefano Gasperini, Nassir Navab, and Federico Tombari. Supergseg: Open-vocabulary 3d segmentation with structured super-gaussians. *arXiv preprint arXiv:2412.10231*, 2024. 1, 3, 4, 5, 6

[22] Shilong Liu, Zhaoyang Zeng, Tianhe Ren, Feng Li, Hao Zhang, Jie Yang, Qing Jiang, Chunyuan Li, Jianwei Yang, Hang Su, et al. Grounding dino: Marrying dino with grounded pre-training for open-set object detection. In *European conference on computer vision*, pages 38–55. Springer, 2024. 2

[23] Tao Lu, Mulin Yu, Linning Xu, Yuanbo Xiangli, Limin Wang, Dahua Lin, and Bo Dai. Scaffold-gs: Structured 3d gaussians for view-adaptive rendering. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 20654–20664, 2024. 1, 3, 4

[24] Dominic Maggio, Yun Chang, Nathan Hughes, Matthew Trang, Dan Griffith, Carlyn Dougherty, Eric Cristofalo,

Lukas Schmid, and Luca Carlone. Clio: Real-time task-driven open-set 3d scene graphs. *IEEE Robotics and Automation Letters*, 2024. 1

[25] 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. *Communications of the ACM*, 65(1):99–106, 2021. 1

[26] Thomas Müller, Alex Evans, Christoph Schied, and Alexander Keller. Instant neural graphics primitives with a multiresolution hash encoding. *ACM transactions on graphics (TOG)*, 41(4):1–15, 2022. 1

[27] Michael Niemeyer, Fabian Manhardt, Marie-Julie Rakotosaona, Michael Oechsle, Daniel Duckworth, Rama Gosula, Keisuke Tateno, John Bates, Dominik Kaeser, and Federico Tombari. Radsplat: Radiance field-informed gaussian splatting for robust real-time rendering with 900+ fps. In *International Conference on 3D Vision 2025*. 1

[28] Maxime Oquab, Timothée Darcet, Théo Moutakanni, Huy Vo, Marc Szafraniec, Vasil Khalidov, Pierre Fernandez, Daniel Haziza, Francisco Massa, Alaaeldin El-Nouby, et al. Dinov2: Learning robust visual features without supervision. *Transactions on Machine Learning Research Journal*, 2024. 2

[29] Songyou Peng, Kyle Genova, Chiyu Jiang, Andrea Tagliasacchi, Marc Pollefeys, Thomas 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. 2

[30] Yuning Peng, Haiping Wang, Yuan Liu, Chenglu Wen, Zhen Dong, and Bisheng Yang. Gags: Granularity-aware feature distillation for language gaussian splatting. *arXiv preprint arXiv:2412.13654*, 2024. 6

[31] Minghan Qin, Wanhua Li, Jiawei Zhou, Haoqian Wang, and Hanspeter Pfister. Langsplat: 3d language gaussian splatting. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 20051–20060, 2024. 1, 2, 3, 4, 5, 6, 7, 8

[32] Yansong Qu, Shaohui Dai, Xinyang Li, Jianghang Lin, Liujuan Cao, Shengchuan Zhang, and Rongrong Ji. Goi: Find 3d gaussians of interest with an optimizable open-vocabulary semantic-space hyperplane. In *Proceedings of the 32nd ACM international conference on multimedia*, pages 5328–5337, 2024. 1, 3, 5, 6

[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. PmLR, 2021. 2, 6, 1

[34] Philipp A Rauschnabel, Reto Felix, Chris Hinsch, Hamza Shahab, and Florian Alt. What is xr? towards a framework for augmented and virtual reality. *Computers in human behavior*, 133:107289, 2022. 1

[35] Tianhe Ren, Shilong Liu, Ailing Zeng, Jing Lin, Kunchang Li, He Cao, Jiayu Chen, Xinyu Huang, Yukang Chen, Feng Yan, et al. Grounded sam: Assembling open-world models for diverse visual tasks. *arXiv preprint arXiv:2401.14159*, 2024. 2

[36] Yunhang Shen, Chaoyou Fu, Peixian Chen, Mengdan Zhang, Ke Li, Xing Sun, Yunsheng Wu, Shaohui Lin, and Rongrong Ji. Aligning and prompting everything all at once for universal visual perception. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 13193–13203, 2024. 2

[37] Jin-Chuan Shi, Miao Wang, Hao-Bin Duan, and Shao-Hua 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. 5, 6

[38] Lei Tian, Xiaomin Li, Liqian Ma, Hefei Huang, Zirui Zheng, Hao Yin, Taiqing Li, Huchuan Lu, and Xu Jia. Ccl-lgs: Contrastive codebook learning for 3d language gaussian splatting. *arXiv preprint arXiv:2505.20469*, 2025. 3

[39] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. *Advances in neural information processing systems*, 30, 2017. 5

[40] Xiaofeng Wang, Zheng Zhu, Guan Huang, Xinze Chen, Jiagang Zhu, and Jiwen Lu. Drivedreamer: Towards real-world-drive world models for autonomous driving. In *European conference on computer vision*, pages 55–72. Springer, 2024. 1

[41] Yuqi Wang, Jiawei He, Lue Fan, Hongxin Li, Yuntao Chen, and Zhaoxiang Zhang. Driving into the future: Multiview visual forecasting and planning with world model for autonomous driving. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 14749–14759, 2024. 1

[42] Yanmin Wu, Jiarui Meng, Haijie Li, Chenming Wu, Yahao Shi, Xinhua Cheng, Chen Zhao, Haocheng Feng, Errui Ding, Jingdong Wang, et al. Opengaussian: Towards point-level 3d gaussian-based open vocabulary understanding. *Advances in Neural Information Processing Systems*, 37:19114–19138, 2024. 2, 3, 4, 5, 6, 7, 8, 1

[43] Qiangeng Xu, Zexiang Xu, Julien Philip, Sai Bi, Zhixin Shu, Kalyan Sunkavalli, and Ulrich Neumann. Point-nerf: Point-based neural radiance fields. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 5438–5448, 2022. 1

[44] Mingqiao Ye, Martin Danelljan, Fisher Yu, and Lei Ke. Gaussian grouping: Segment and edit anything in 3d scenes. In *European conference on computer vision*, pages 162–179. Springer, 2024. 5, 6

[45] Haiyang Ying, Yixuan Yin, Jinzhi Zhang, Fan Wang, Tao Yu, Ruqi Huang, and Lu Fang. Omniseg3d: Omniversal 3d segmentation via hierarchical contrastive learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 20612–20622, 2024. 1, 2, 4

[46] Zehao Yu, Anpei Chen, Binbin Huang, Torsten Sattler, and Andreas Geiger. Mip-splatting: Alias-free 3d gaussian splatting. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 19447–19456, 2024. 1

[47] 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 *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 21676–21685, 2024. 1, 2, 3, 4, 5, 6