2026 IEEE International Conference on Robotics and Automation (ICRA 2026)
June 1-5, 2026. Vienna, Austria

Semantically Consistent Language Gaussian Splatting for 3D Point-Level Open-vocabulary Querying

Hairong Yin[1], Huangying Zhan[2], Yi Xu[2], Raymond A. Yeh[1]

***Abstract*—Open-vocabulary 3D scene understanding is crucial for robotics applications, such as natural language-driven manipulation, human-robot interaction, and autonomous navigation. Existing methods for querying 3D Gaussian Splatting often struggle with inconsistent 2D mask supervision and lack a robust 3D point-level retrieval mechanism. In this work, (i) we present a novel point-level querying framework that performs tracking on segmentation masks to establish a semantically consistent ground-truth for distilling the language Gaussians; (ii) we introduce a GT-anchored querying approach that first retrieves the distilled ground-truth and subsequently uses the ground-truth to query the individual Gaussians. Extensive experiments on three benchmark datasets demonstrate that the proposed method outperforms state-of-the-art performance. Our method achieves an mIoU improvement of +4.14, +20.42, and +1.7 on the LERF, 3D-OVS, and Replica datasets. These results validate our framework as a promising step toward open-vocabulary understanding in real-world robotic systems.**

I. Introduction

Querying open-vocabulary objects in 3D scenes, *i.e.*, identifying and isolating scene components based on natural language descriptions, is a fundamental capability necessary to advance robotic perception and interaction. To evaluate open-vocabulary querying, recent works [11, 16, 21, 26, 31, 35] have formulated the task in two steps: **(a)** creating a 3D scene representation augmented with language-aligned features, and **(b)** querying this representation effectively using language. Existing works address **(a)** by distilling language embeddings from foundation 2D vision-language models (VLMs) [15, 22] into point clouds [8, 28], NeRF [19, 20, 27] and 3D Gaussians [3, 10, 32]. To address **(b)**, they perform retrieval by thresholding on the cosine similarity between the text query embedding and the distilled language embedding.

While many methods have explored this task, a distinction lies in their output querying format. Approaches, *e.g.*, LangSplat [21], that produce *2D segmentation masks* are insufficient for robotics, as downstream tasks like motion planning, grasp synthesis, and collision avoidance often operate directly on a 3D representation. This led to the development of **point-level querying** [16, 26], which directly retrieves a subset of the underlying *3D primitives*. In the case of 3D Gaussian Splatting [10], this involves retrieving the relevant subset of Gaussians from the scene. The resulting 3D selection provides a direct and actionable representation for a robot's planning and control.

[1]Department of CS, Purdue University, USA. {yin178, rayyeh}@purdue.edu.
[2]Goertek Alpha Labs, USA. {huangying.zhan, yi.xu}@goertekusa.com.
Paper website: *https://evelinyin.github.io/seconGS/*

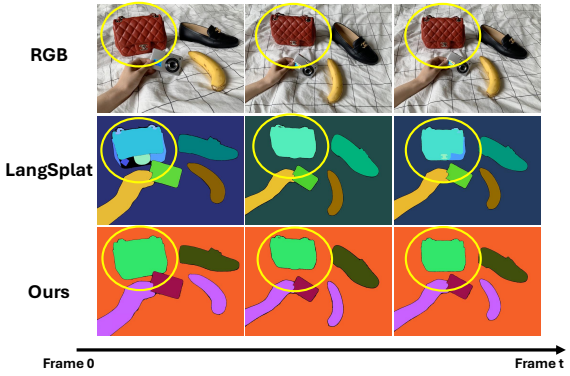

Fig. 1: Visualization of the language embedding supervision. For the "red bag" circled in yellow, the ground-truth constructed by LangSplat [21] is inconsistent across frames, while Ours remains consistent across frames.

In this work, we identify two main shortcomings of LangSplat [21] (initially proposed for 2D querying) when applied to 3D point-level querying. First, we observe inconsistency in the distillation ground-truth language embeddings constructed by LangSplat. That is, the embeddings are different for the same object instance across different frames; see in Fig. 1. To address this, we propose a tracking-based distillation process and aggregate the language embedding into a consistent ground-truth.

Next, the second challenge lies in the querying phase. LangSplats' querying approach thresholds the similarity between query text vectors and learned point-wise language embeddings; however, choosing an appropriate threshold across different text queries is challenging. As shown in Fig. 2, the optimal thresholds are *not the same* across all objects. To mitigate this difficulty, we propose a novel Ground-Truth Anchored (GT-Anchored) querying method, which computes the threshold relative to, "anchored", ground-truth (GT) used in the distillation process instead of directly with the text query.

Empirically, we conduct experiments over three datasets: LERF [11], 3D-OVS [17], and Replica [25], demonstrating that our method outperforms the state-of-the-art method in terms of mIoU by $+4.14$, $+20.42$, and $+1.74$, respectively. A detailed ablation study is conducted to verify the effectiveness of the proposed components.

Our contributions are as follows:

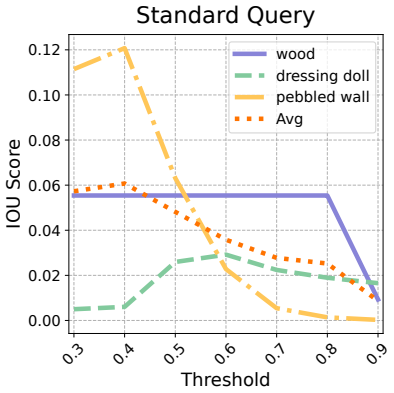

Fig. 2: IoU metric per query vs. cosine similarity thresholds for the standard querying method. We observe that it does not have a consistent optimal threshold for all queries.

- We introduce tracking for generating semantic and 3D-consistent ground-truth to train language-aware Gaussians, which improves the distillation quality.
- With this improved 3D language Gaussians, we propose an effective GT-anchored querying process by leveraging the created consistent ground-truth to alleviate the aforementioned challenge of selecting a suitable threshold.
- Extensive experiments across three datasets demonstrate that our approach outperforms existing open-vocabulary 3D querying methods.

II. Related Work

A. Vision foundation models.

Several open-sourced foundation models [1, 13, 22, 23] have become the bedrock of many works [2, 8]. These foundation models' capabilities can either be used directly or their features can be distilled into another model. In language and vision, CLIP [22] is a model that is capable of encoding images and natural language text to the same embedding space. Using this joint embedding space, they demonstrate zero-shot capability for image classification, which is later generalized to segmentation [34] and language segmentation (LSeg) [15].

In the area of image segmentation, the Segment Anything Model (SAM) [13] is a notable foundation model. SAM's segmentation capabilities have been adapted and extended to 3D tasks. Recent works [2, 12] have leveraged SAM to integrate semantic information into NeRFs, enabling the extraction of 3D masks for target objects. Other approaches [7, 18] have incorporated semantic features into point clouds, enhancing object representation and segmentation in 3D. The process in 3D Gaussian Splatting [10] has further motivated studies [9, 31] that focus on object representation in both 3D and 4D, including advancements in interactive segmentation and object tracking. Recently, SAM2 [23] extended SAM's capability to tracking of masklets, *i.e.*, consistent masks across both space and time.

B. Open-vocabulary 3D scene understanding.

With advancements in 3D scene representation, there is a surge in interest in incorporating semantics/language into 3D representation. LERF [11] and other works [17, 24, 30] distilled features from DINO [1] and CLIP [22] to learn a NeRF [20], or leveraged 2D annotations [33] to construct feature/language fields. Others [8, 28, 29] distill knowledge from these language-rich models into point clouds or voxels, enabling open-vocabulary 3D scene understanding. In more recent works [6, 21, 26, 31, 35], there is a shift towards 3D Gaussian Splatting [10].

More closely related to our work is LangSplat [21], which augments 3D scenes with language features distilled from CLIP, enabling natural language querying on the renderings of the 3D scene. Gaussian Grouping [31] jointly performs 3D reconstruction and segmentation of open-world objects. It generates 2D masks using SAM and associates them across frames through a zero-shot tracker. The method also incorporates a custom loss that enforces 3D consistency. However, the method leverages the tracking information differently from ours by learning a group ID for each Gaussian point, which relies on a complicated grouping loss. Also, it selects target objects using only the semantics of the first frame, which could lead to potential query failures. OpenGaussian [26] introduces new loss functions that leverage inter- and intra-mask smoothness relationships, along with a codebook-based clustering method to improve instance-level association of 3D points.

Differently, our approach does not rely on new loss functions. Instead, our tracking-based method extracts masklets to construct consistent ground-truth supervision and introduces a novel GT-anchored querying procedure.

III. Preliminaries

We review LangSplat and introduce the necessary notation. **LangSplat** [21] represents a 3D scene with a set of 3D Gaussians $\mathcal{G} = \{\boldsymbol{g}_i\}$, where each Gaussian $\boldsymbol{g}_i$ is associated with the parameters

$$\boldsymbol{g}_i \triangleq (\boldsymbol{\mu}_i, \boldsymbol{\Sigma}_i, \boldsymbol{c}_i, \alpha_i, \boldsymbol{l}_i) \tag{1}$$

corresponding to the 3D location, covariance matrix, color, opacity, and a language embedding. Different from a regular 3D Gaussian splatting [10], each of the Gaussians (Eq. (1)) includes a language embedding $\boldsymbol{l}_i \in \mathbb{R}^D$ to encode the semantics of a 3D scene.

This language embedding can then be rendered into a language field $\hat{\boldsymbol{L}}_\pi \in \mathbb{R}^{H \times W \times D}$, where H and W correspond to the height and width of the rendered image at a camera pose π. This is done by using a tile-based rasterization, just as one would for colors. The language feature at each pixel $\boldsymbol{p}$ is computed as

$$\hat{\boldsymbol{L}}_\pi[\boldsymbol{p}] = \sum_{i \in \mathcal{T}} \boldsymbol{l}_i f_i^{\text{2D}}(\boldsymbol{p}) \prod_{j=1}^{i-1} (1 - f_j^{\text{2D}}(\boldsymbol{p})), \tag{2}$$

where f_i^{2D} represents the the projected 2D contribution of a 3D Gaussian $\boldsymbol{g}_i$, and $\mathcal{T}$ is the set of Gaussians in a tile.

LangSplat training. Let $\hat{\boldsymbol{L}}_{\pi_t}$ denote the rendering of the scene from the camera pose π_t associated with image $\boldsymbol{I}_t$. LangSplat trains language embedding $\boldsymbol{l}_i$ by minimizing the L1

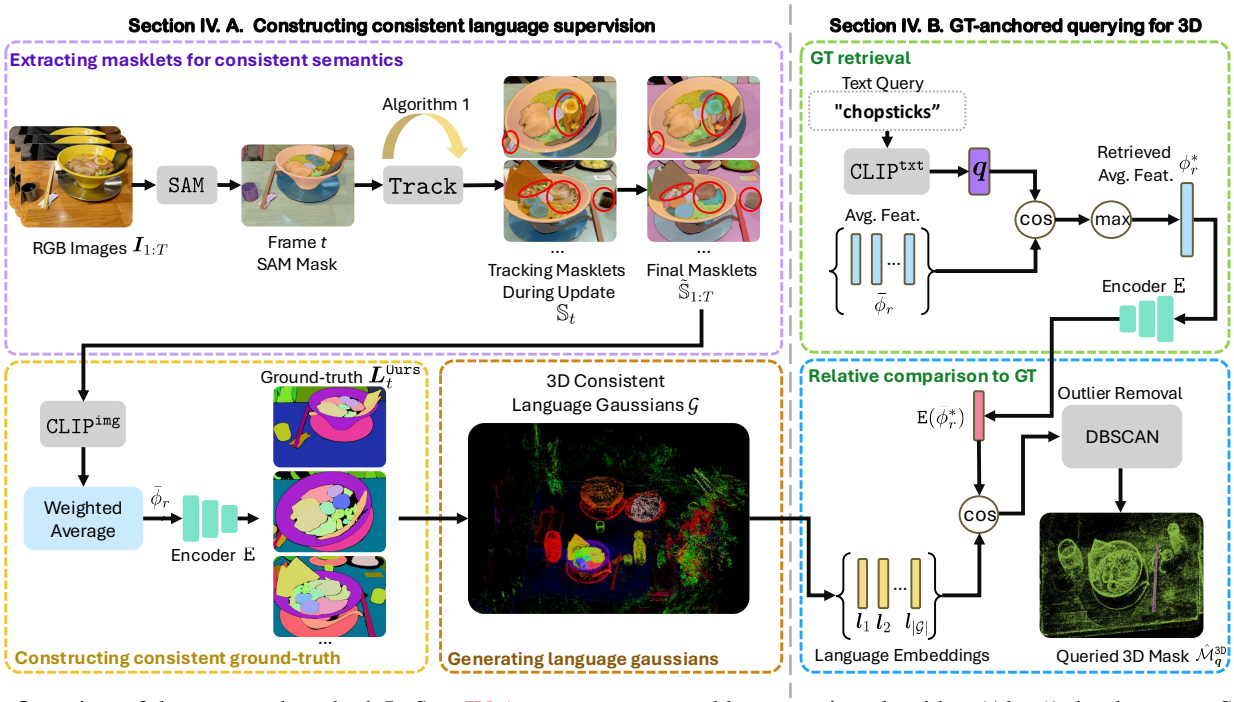

Fig. 3: Overview of the proposed method. In Sec. IV-A, we present a masklet extraction algorithm (Alg. 1) that leverages Segment Anything Models to generate consistent ground-truth $\boldsymbol{L}_t^{\texttt{Ours}}$ for training the language parameters $\boldsymbol{l}_1, \ldots, \boldsymbol{l}_{|\mathcal{G}|}$. In Sec. IV-B, we discuss the GT-anchored retrieval procedure. Rather than directly querying the language parameters $\boldsymbol{l}_i$ with the query vector $\boldsymbol{q}$, we first retrieve the features $\bar{\phi}_r$ that are used to construct the ground truth $\boldsymbol{L}_t^{\texttt{Ours}}$. Then we query the 3D language Gaussian using the encoded feature $\texttt{E}(\bar{\phi}_r^*)$, followed by an outlier removal using DBSCAN [5] to obtain the final result.

loss between the rendering of the language feature $\hat{\boldsymbol{L}}_\pi$ and a "ground-truth" language feature $\boldsymbol{L}_t$, *i.e.*,

$$\min_{\{\boldsymbol{l}_i \forall i\}} \mathbb{E}_{\boldsymbol{I}_t} \left[\text{L1}(\hat{\boldsymbol{L}}_{\pi_t}, \boldsymbol{L}_t) \right]. \tag{3}$$

Importantly, how one designs this "ground truth" significantly affects the query performance.

In LangSplat, the ground-truth is distilled from a pretrained CLIP [4] with the help of the segment anything model (SAM) [13]. Given an RGB image $\boldsymbol{I}$, SAM extracts a set of non-overlapping segmentation masks $\mathbb{S}_{\boldsymbol{I}} \triangleq \{\mathcal{S}_r\} = \texttt{SAM}(\boldsymbol{I})$, where each $\mathcal{S}_r$ corresponds to the segmented mask of a region r. Here, we broadly use the term "region" to mean a set of related pixels, *e.g.*, an object or a subpart of an object.

Next, a CLIP embedding $\phi_{r,t}$ is extracted for each region at time t, LangSplat [21] masks the image, then passes it to CLIP's image encoder $\texttt{CLIP}^{\text{img}} : \mathbb{R}^{H \times W \times 3} \to \mathbb{R}^D$, *i.e.*,

$$\phi_{r,t} \triangleq \texttt{CLIP}^{\text{img}}(\boldsymbol{I}_t \odot \boldsymbol{S}_r), \tag{4}$$

where $\boldsymbol{S}_r$ corresponds to the mask $\mathcal{S}_r$ in matrix form, and $\odot$ denotes an element-wise multiplication. These extracted region features are then placed back to their respective regions to form the ground-truth

$$\boldsymbol{L}_t[(h,w)] \triangleq \phi_{r,t} \text{ if } (h,w) \in \mathcal{S}_r \ \forall r. \tag{5}$$

As the language embedding $\boldsymbol{L}$ is distilled from a pretrained CLIP embedding, the query vector $\boldsymbol{q}$ is extracted from the pretrained CLIP text encoder. This ensures that both the query vector and the 3D language embeddings are in the same space, allowing for retrieval.

Lastly, we note an implementation detail, LangSplat [21] trains an autoencoder, consisting of an encoder $\texttt{E}$ and a decoder $\texttt{D}$, to reduce the dimensions of the CLIP features, where $\texttt{E} \circ \texttt{D}$ approximates an identity function. With this autoencoder, all the aforementioned formulations can be done in a lower-dimensional space to save GPU memory. However, this dimensionality reduction introduces a trade-off: language features for the same object become less consistent across views due to compression, as illustrated in Fig. 1.

Open-vocabulary 3D (point-level) querying. OpenGaussian [26] proposes to directly query the 3D Gaussians with natural language. Formally, given a trained 3D scene $\mathcal{G}$ and a query $\boldsymbol{q}$, the task is to predict a *"3D mask"*

$$\mathcal{M}_{\boldsymbol{q}}^{\text{3D}} = \{\boldsymbol{g}_i\} \subseteq \mathcal{G}, \tag{6}$$

that indicates whether each Gaussian $\boldsymbol{g}_i$ is relevant to a text query $\boldsymbol{q}$. In other words, a *point-level* querying on the 3D scene's representation rather than on a *rendered image* of a 3D scene.

Tracking framework. Given a sequence of frames $\boldsymbol{I}_{t_1:t_2}$, and a set of candidates $\tilde{\mathcal{B}}_{t_1}$ indicating the regions in frame t_1 that we wish to track and segment, a tracking model extracts a set $\tilde{\mathbb{S}}_{\boldsymbol{I}_{t_1:t_2}}$ of non-overlapping masks across both *space and time*, *i.e.*, "masklets"

$$\tilde{\mathbb{S}}_{\boldsymbol{I}_{t_1:t_2}} \triangleq \{\tilde{\mathcal{S}}_r\} = \texttt{Track}(\boldsymbol{I}_{t_1:t_2}, \mathcal{B}_{t_1}), \tag{7}$$

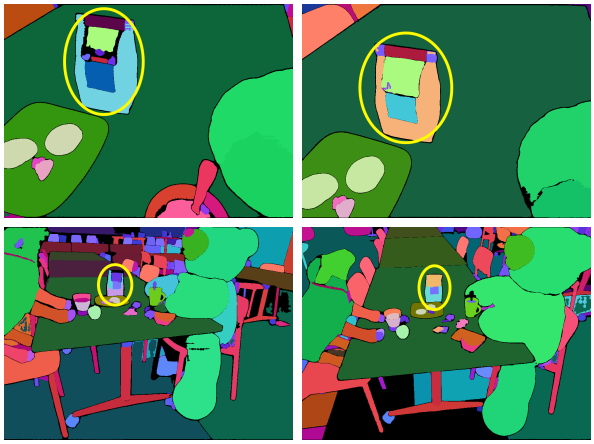

Fig. 4: Visualization of the ground-truth $\boldsymbol{L}_t$ constructed by LangSplat [21]. We observed that the semantics are not consistent across viewpoints, *e.g.*, the circled bag of coffee.

where each $\tilde{\mathcal{S}}_r$ is a set containing voxels (t, h, w) that are associated with the region r. Following the same syntax for a 2D binary mask, we use the tensor $\tilde{\boldsymbol{S}}_r \in \{0,1\}^{T\times H\times W}$ to represent the masklet $\tilde{\mathcal{S}}_r$ in set notation.

IV. Method

Following LangSplat, we use a set of 3D Gaussians augmented with language embeddings to represent a scene. Differently, we propose a novel method for constructing ground-truths that are more semantically consistent and robust across various 3D viewpoints (Sec. IV-A). This approach helps to train better language embeddings for querying. We then introduce a querying method tailored for our learned embeddings (Sec. IV-B). See Fig. 3 for a visual overview.

A. Constructing consistent language supervision

Given a sequence of frames $[\boldsymbol{I}_1, \dots \boldsymbol{I}_T]$ with camera poses, we aim to construct a better ground-truth feature $\boldsymbol{L}_t^{\texttt{Ours}}$ for each of the frames to train LangSplat's parameters by minimizing the objective function in Eq. (3). This improved ground-truth is obtained by ensuring that all pixels within the same region, as identified by the chosen tracking model (SAM2 [23]), share the same CLIP embedding. In other words, the supervision will be semantically consistent up to the quality of the extracted masklets from SAM2.

Inconsistent semantics from LangSplat. The main shortcoming of the ground-truth feature $\boldsymbol{L}_t$ created by LangSplat is its potential inconsistency across different views. In Fig. 4, we visualize these features constructed by LangSplat, where similar colors indicate higher feature similarity. As shown, the bag of cookies (circled in yellow) exhibits significant mask variations across the views, indicating inconsistent supervision. This inconsistency arises because SAM segments each image independently, potentially selecting different regions. Hence, the ground-truth features derived from these segmentations are also inconsistent.

To address this inconsistency, we construct the ground truth $\boldsymbol{L}_t^{\texttt{Ours}}$ by additionally leveraging the tracking capabilities of SAM2. Importantly, we aim to design $\boldsymbol{L}_t^{\texttt{Ours}} \in \mathbb{R}^{H\times W\times D}$ such that each pixel's feature remains consistent across frames when it belongs to the same region extracted by SAM2.

Algorithm 1 Extracting regions with SAM and Tracking

Input: Image sequence $\boldsymbol{I}_{1:T}$, segmentors `SAM` and Tracker `Track`, threshold κ
$\tilde{\mathbb{S}}_{1:T} \leftarrow \{\}$ `# Tracked masklets`
for $t \in \{1, \dots, T\}$ **do**
  $\mathbb{S}_{\boldsymbol{I}_t} = \texttt{SAM}(\boldsymbol{I}_t)$
  `# Check if tracked.`
  for $\boldsymbol{S}_r \in \mathbb{S}_{\boldsymbol{I}_t}$ **do**
    for $\tilde{\boldsymbol{S}}_{r'} \in \tilde{\mathbb{S}}_{1:T}$ **do**
      if $\texttt{IoU}(\tilde{\boldsymbol{S}}_{r'}[t-1], \boldsymbol{S}_r) > \kappa$ **then**
        $\mathbb{S}_{\boldsymbol{I}_t} \leftarrow \mathbb{S}_{\boldsymbol{I}_t} \backslash \{\boldsymbol{S}_r\}$
      end if
    end for
  end for
  `# Adding untracked masklets`
  $\mathcal{B} \leftarrow \texttt{RegionFromMask}(\mathbb{S}_{\boldsymbol{I}_t})$
  $\tilde{\mathbb{S}}_{1:T} \leftarrow \tilde{\mathbb{S}}_{1:T} \cup \texttt{Track}(\boldsymbol{I}_{1:T}, \mathcal{B})$
end for
Output: $\tilde{\mathbb{S}}_{1:T}$

Extracting masklets for consistent semantics. Reviewed in Sec. III, a tracking module takes a sequence of images and regions of interest as input to track masks of the same region. In LangSplat, SAM is used to extract the initial candidates, *i.e.*, for each frame $\boldsymbol{I}_t$, SAM proposes a set of regions $\mathbb{S}_{\boldsymbol{I}_t}$. Starting from the first frame, we check if a proposed region has already been tracked by comparing the mIoU of the SAM mask with the tracked masks and applying a threshold. If the proposed region has not been tracked, we run the tracking model and add the output masklets to the set of tracked masklets $\tilde{\mathbb{S}}_{1:T}$. These steps are summarized in Alg. 1.

Constructing consistent ground-truth. With the set of masklets $\tilde{\mathbb{S}}_{1:T}$ extracted, we create a consistent ground-truth by aggregating the CLIP embedding $\bar{\phi}_r$ for each masklet $\tilde{\mathcal{S}}_r \in \tilde{\mathbb{S}}_{1:T}$. This is done by masking out the image $\boldsymbol{I}_t$ using the extracted masklet and then passing it to CLIP's image encoder:

$$\bar{\phi}_r = \sum_{t=1}^{T} \omega_t \cdot \texttt{CLIP}^{\texttt{img}}(\boldsymbol{I}_t \odot \tilde{\boldsymbol{S}}_r[t]), \tag{8}$$

where $\tilde{\boldsymbol{S}}_r$ denotes the masklet $\tilde{\mathcal{S}}_r$ represented in a tensor, and ω_t denotes the ratio of pixels in $\tilde{\boldsymbol{S}}_r[t]$ to the total pixel count in $\tilde{\boldsymbol{S}}_r$. In other words, the embedding from each view is weighted proportionally to the number of pixels in the segmentation.

The main intuition behind this design is that averaging reduces variance. The proposed $\bar{\phi}_r$ is more consistent than the individual $\phi_{r,t}$ used in LangSplat when used as supervision for the distillation. Furthermore, the weighting scheme helps to suppress the contribution of small regions that often contain noisier language embeddings, *i.e.*, we consider the reliability of individual features.

TABLE I: Quantitative results on LERF. For OpenGaussian, we report the numbers from their paper.

Methods	mIoU↑					mAcc↑					Loc. Acc↑				
	figurines	ramen	teatime	kitchen	**Avg**	figurines	ramen	teatime	kitchen	**Avg**	figurines	ramen	teatime	kitchen	**Avg**
LangSplat-m [21]	12.43	6.39	20.60	17.58	14.25	21.43	7.04	37.29	18.18	20.99	5.36	0.00	3.39	4.55	3.33
GSGroup.-m [31]	7.75	8.80	10.94	16.29	10.95	10.71	9.86	10.17	27.27	14.50	10.71	2.82	5.08	4.55	5.79
OpenGauss. [26]	39.29	31.01	**60.44**	22.7	38.36	55.36	42.25	**76.27**	31.82	51.43	-	-	-	-	-
Ours	**58.91**	**37.85**	43.57	**29.67**	**42.50**	**82.14**	**61.97**	54.24	**50.00**	**62.09**	**82.14**	**61.97**	**62.71**	**40.91**	**61.93**

TABLE II: Quantitative results on the 3D-OVS dataset.

Methods	mIoU↑						mAcc↑						Loc. Acc↑					
	bed	bench	lawn	room	sofa	**Avg**	bed	bench	lawn	room	sofa	**Avg**	bed	bench	lawn	room	sofa	**Avg**
LangSplat-m [21]	29.83	17.38	33.64	23.35	24.64	25.77	43.33	28.57	63.33	33.33	43.33	42.38	3.00	48.57	40.00	43.33	43.33	35.64
GSGroup.-m [31]	48.51	35.49	65.13	46.39	29.86	45.08	**100.00**	71.43	**100.00**	76.67	40.00	77.62	**96.67**	42.86	**100.00**	53.33	43.33	67.24
OpenGauss. [26]	48.50	46.02	64.63	47.60	44.06	50.16	**100.00**	57.14	**100.00**	**83.33**	**66.67**	81.43	23.33	37.14	20.00	50.00	56.67	37.43
Ours	**56.81**	**87.58**	**87.12**	**64.70**	**56.70**	**70.58**	66.67	**100.00**	**100.00**	**83.33**	**66.67**	**83.33**	83.33	**100.00**	**100.00**	**83.33**	**100.00**	**93.33**

TABLE III: Quantitative results on Replica dataset. To compute mIoU and mAcc using the ground-truth point clouds, we skip the densification stage when training 3D Gaussian Splatting for all methods.

Methods	mIoU↑									mAcc↑								
	office0	office1	office2	office3	office4	room0	room1	room2	**Avg**	office0	office1	office2	office3	office4	room0	room1	room2	**Avg**
LangSplat-m [21]	2.43	2.1	5.68	4.65	1.49	3.86	4.08	0.92	3.15	11.09	1.36	10.7	13.99	2.37	12.82	12.24	10.05	9.33
GSGroup.-m [31]	19.58	0.00	32.77	10.18	30.29	13.08	17.81	17.06	17.60	38.42	0.00	**74.48**	26.17	45.67	36.21	31.57	24.17	34.59
OpenGauss. [26]	17.20	**23.13**	**43.72**	**42.36**	61.33	31.45	40.36	42.14	37.71	36.54	35.11	66.38	42.64	69.62	41.74	31.72	54.01	47.22
Ours	**25.77**	20.15	15.06	37.29	**64.83**	**40.33**	**64.39**	**47.81**	**39.45**	**50.76**	**35.97**	29.01	**45.12**	**82.85**	**60.00**	**84.72**	**63.64**	**56.51**

To construct the ground truth $\boldsymbol{L}_t^{\texttt{Ours}}$ for a frame $\boldsymbol{I}_t$, we place the averaged CLIP embedding into the pixel location of each frame according to the masklet. For all extracted masklets $\tilde{\mathcal{S}}_r \in \tilde{\mathcal{S}}_{1:T}$,

$$\boldsymbol{L}_t^{\texttt{Ours}}[(h,w)] = \bar{\phi}_r \text{ if } (t,h,w) \in \tilde{\mathcal{S}}_r. \tag{9}$$

For pixels across views that are tracked and segmented into the same region, this construction of ground-truth assigns the same averaged CLIP embedding. Compared with LangSplat's ground-truth in Eq. (5), our construction is **shared across time** t, *i.e.*, the language embedding $\boldsymbol{l}_i$ in LangSplat receives consistent supervision across all relevant frames.

Autoencoder details. As in LangSplat, to reduce the GPU memory usage, we train a light-weight autoencoder consisting of an encoder $\texttt{E}$ and a decoder $\texttt{D}$. Differently, we encode the averaged CLIP embedding $\bar{\phi}_r$ into a low-dimensional latent space, *i.e.*,$\texttt{E}(\bar{\phi}_r)$ is used as supervision to train the language embeddings $\boldsymbol{l}_i$. A side note: as the autoencoder is trained using our constructed, more consistent, averaged CLIP embedding, it also has an easier job in learning the reconstruction.

B. Ground Truth (GT)-anchored 3D Querying

With the text query vector $\boldsymbol{q}$, the standard approach is to directly compares the CLIP features $\boldsymbol{q}$ of the query text with the language embeddings $\boldsymbol{l}_i$ of each Gaussian, *i.e.*

$$\hat{\mathcal{M}}_{\boldsymbol{q}}^{\texttt{3D,1step}} \triangleq \{\boldsymbol{g}_i |\ \forall i\ \texttt{Cos}(\boldsymbol{l}_i, \boldsymbol{q}) \geq \text{threshold}\}, \tag{10}$$

where $\texttt{Cos}$ is the cosine similarity. However, this one-step approach struggles to find a single effective threshold across different language embeddings $\boldsymbol{l}_i$ as CLIP's image and language embeddings are known to be not well calibrated [14]. To address this challenge, we propose our GT-anchored approach for querying 3D Gaussians. This involves first retrieving the ground truth and then comparing the similarity relative to the ground truth, as described in more detail below.

GT retrieval. Given the CLIP feature of a text query $\boldsymbol{q} \in \mathbb{R}^{512}$, we first apply a low threshold to filter out invalid prompts. We then retrieve the most similar average feature (GT for distillation) over all regions feature

$$\bar{\phi}_r^* \triangleq \underset{r \in \{r' | \texttt{Cos}(\bar{\phi}'_{r'}, \boldsymbol{q}) \geq \text{threshold}\}}{\arg\max} \texttt{Cos}(\bar{\phi}_r, \boldsymbol{q}). \tag{11}$$

As $\bar{\phi}_r$ is obtained as a weighted average of CLIP image embeddings and $\boldsymbol{q}$ comes from CLIP text embeddings, a direct comparison between them through cosine similarity is effective and does not involve a threshold.

Relative comparison to GT. With the retrieved GT $\bar{\phi}_r^*$, we compress it into lower dimension with the pretrained encoder $\texttt{E}$. Then we compute its cosine similarity with the learned language embedding $\boldsymbol{l}_i$ for each Gaussian. Next, we threshold this similarity to retrieve the tentative set of Gaussians

$$\tilde{\mathcal{M}}_{\boldsymbol{q}}^{\texttt{3D}} = \{\boldsymbol{g}_i |\ \forall i\ \text{Cos}(\boldsymbol{l}_i, \texttt{E}(\bar{\phi}_r^*)) \geq \text{threshold}\}, \tag{12}$$

where the querying process compares $\texttt{E}(\bar{\phi}_r^*)$ with $\boldsymbol{l}_i$. Recall $\texttt{E}(\bar{\phi}_r^*)$ is used as supervision for training language embeddings $\boldsymbol{l}_i$, *i.e.*, the relevant language embeddings are trained to be similar to $\texttt{E}(\bar{\phi}_r^*)$. Therefore, any high threshold works well, which improves the queries' reliability and robustness.

V. Experiments

Datasets. Following LangSplat [21], we conduct experiments on the further annotated LERF [11] dataset that contains a set of in-the-wild scenes and on the 3D-OVS [17] dataset, which includes a collection of long-tail objects for evaluating open-vocabulary 3D querying. Additionally, we report results on the Replica [25] dataset, which has labeled point clouds

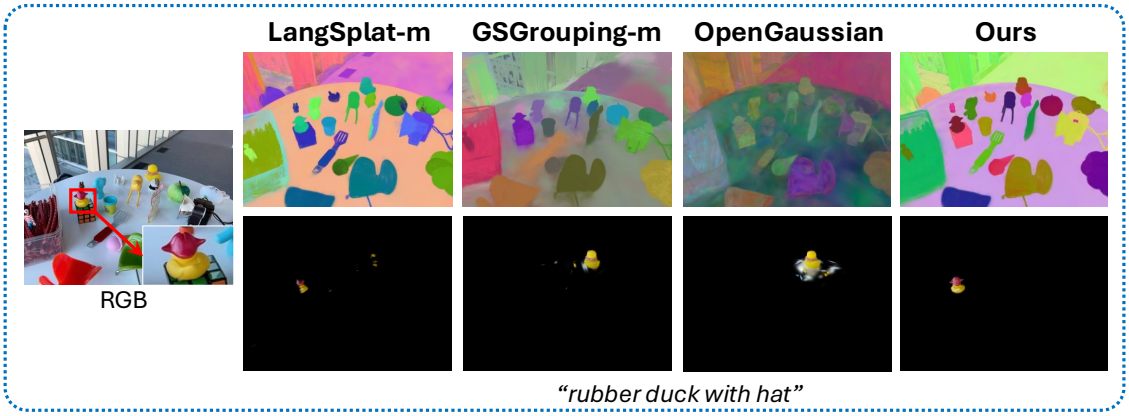

Fig. 5: Qualitative results on LERF dataset of scene"ramen" and "figurines". For each scene, the first row contains rendered language embeddings, and the second row contains 3D query results.

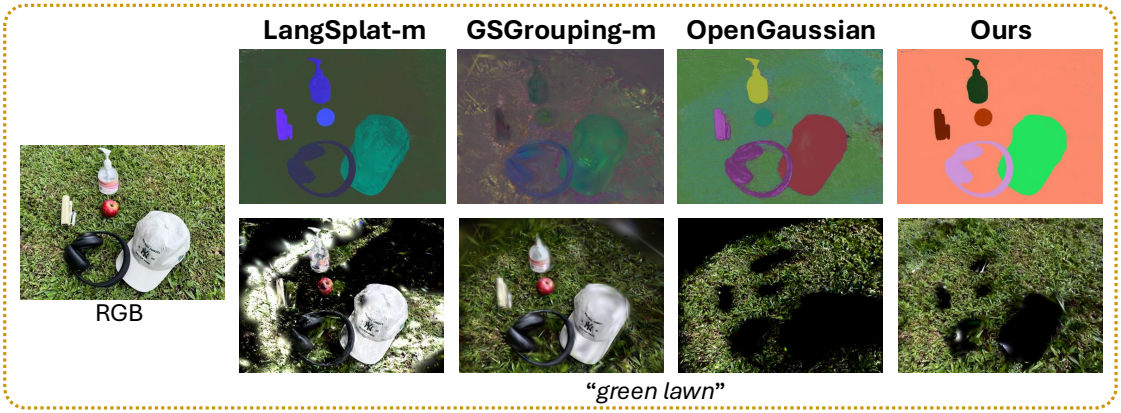

Fig. 6: Qualitative results on 3D-OVS dataset for scene "lawn". The first row contains rendered language embeddings, and the second row contains 3D query results for "green lawn".

for indoor objects. We then evaluate on ten object classes for querying over eight senses in Replica.

Evaluation metrics. As the ground-truths for LERF and 3D-OVS datasets are provided in 2D, we measure the performance of 3D querying indirectly. We render the queried Gaussians in $\hat{\mathcal{M}}_q^{3D}$ to obtain a 2D mask for evaluation using 2D metrics following LangSplat [21]. This includes mean Intersection over Union (mIoU ↑) and localization accuracy (Loc. Acc ↑). Here, mIoU is defined as the ratio of the intersecting pixels to the total number of pixels in the union of the predicted masks and ground-truth masks.

For Loc. Acc, a query is considered correct if the center of the queried mask's exterior bounding box falls within the bounding box of the ground-truth. We also report mIoU accuracy (mAcc↑), a 2D metric proposed by OpenGaussian [26], where a query is considered correct if its IoU is greater than 0.25. For the Replica dataset, which contains 3D retrieval ground-truths, mIoU and mAcc are instead computed on the set of 3D locations of the Gaussians.

Baselines. For a fair comparison, we strictly followed *OpenGaussian* [26] for the task of open-vocabulary 3D (point-level) querying. As they reported on the LERF dataset, we directly included their results from the paper. For 3D-OVS and Replica datasets, we use their publicly released implementation. To further benchmark the performance, we included more baseline methods **modified (m) for** direct 3D querying: *LangSplat-m* [21], which also trains a language 3D Gaussian. Hence, the standard query approach in Eq. (10) can be directly used, following how OpenGaussian [26] evaluates. *GaussianGrouping-m* [31], which we follow their implementation for the open-vocabulary query to select group IDs, and use the corresponding Gaussian points as candidates.

Implementation Details. To extract language features, we use the OpenCLIP ViT-B/16 model. For 2D mask segmentation, we employ the SAM ViT-H model, and for tracking masks of the same object, we utilize the SAM2-hiera-large model. Pretraining the standard 3D Gaussian Splatting takes 30,000 steps. This is followed by training the language embeddings for an additional 30,000 steps, skipping the densification stage. Experiments are conducted on an NVIDIA A100 GPU.

A. *Quantitative results*

LERF dataset. In Tab. I, we show the results on the LERF dataset. We observe that Ours consistently outperforms LangSplat-m and, on average, is better than OpenGaussian, achieving an improvement of +4.14 in mIoU and +10.66 in mAcc. We believe this gain is significant, as retraining the

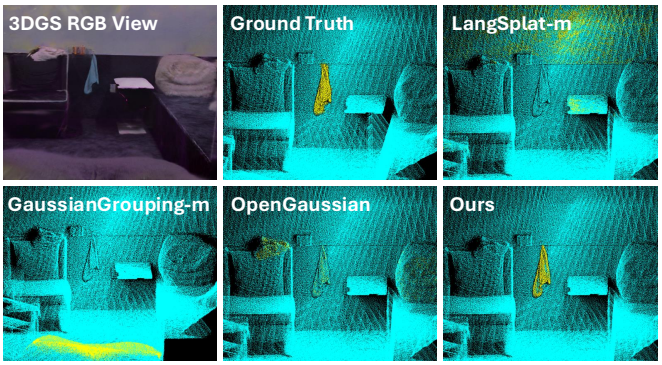

Fig. 7: Qualitative results on Replica dataset. Yellow points are the queried points of *"cloth"*.

language Gaussians 5 times with different seeds has a small standard deviation, *e.g.*, on figurines the std is 0.24. We observe that LangSplat-m and GaussianGrouping-m, methods designed for 2D querying, do not generalize well to point-level querying.

3D-OVS dataset. The results for the 3D-OVS dataset is reported in Tab. II. Our method achieves 70.58% in mIoU and 93.33% in Loc. Acc, significantly outperforming baseline methods. Notably, our method demonstrates a $+20.42$ gain in mIoU averaged across scenes. We observe that there exists 100% in the Loc. Acc because the dataset is relatively easy with ≤ 30 frames and ≤ 7 objects in each scene, leading to effective segmentation and tracking from SAM2. Next, we observed that OpenGaussian has a low Loc. Acc due to small "floaters" in the retrieved Gaussians. These floaters shift the center point of the exterior bounding box to an incorrect location.

Replica dataset. We show results in Tab. III. We observe that our approach outperforms OpenGaussian with an average gain of $+1.74$ on mIoU and $+9.29$ on mAcc. There exists 0.00 in GaussianGrouping-m's results because their query implementation only uses the first frame's semantics. In large scenes like Replica, the query object may not appear initially, causing an empty query.

B. Qualitative results

LERF dataset. To further analyze the methods, we visualize the learned language embeddings and the queried Gaussians in Fig. 5. We observe that baselines produce less consistent language embeddings, evident from the blurriness along object boundaries. Our method achieves cleaner and more fine-grained object localization in 3D space based on text queries, as illustrated in the second row for each scene in Fig. 5. Note that all four methods encounter a common failure mode of empty query, *i.e.*, no valid Gaussians are returned for a text query, resulting in zero IoU for that query.

3D-OVS dataset. Fig. 6 shows the qualitative results on 3D-OVS dataset. Our method creates more consistent language embeddings with crisp object boundaries. Furthermore, our method's retrieval results are less noisy and preserve the complete structure of each queried object, *i.e.*, the lawn is accurately retrieved with query "green lawn".

TABLE IV: Ablation study on each proposed component using LERF's "figurines" scene. Note that combining the language features relies on leveraging tracking information and is not feasible without it. So we put "-" in the first row. The best result is achieved with all our proposed components.

Ablations			Metrics		
Tracking	$\bar{\phi}$ Strategy	GT-anchored	mIoU	mAcc	Loc. Acc
	-	✓	4.56	7.14	5.36
✓	$\bar{\phi}_r$		9.09	12.50	8.93
✓	$\bar{\phi}_r^{\texttt{stand.}}$	✓	48.84	67.86	23.21
✓	$\bar{\phi}_r$	✓	**58.91**	**82.14**	**30.36**

TABLE V: Ablation study on DBSCAN and Canonical Query on scene "figurines" in LERF.

Ablations				Metrics		
SAM2	GT-anchored	DBSCAN	Cano. Query	mIoU	mAcc	Loc. Acc
✓		✓	✓	15.67	26.79	14.29
✓	✓			40.06	66.07	14.50
✓	✓	✓		**58.91**	**82.14**	**30.36**

Replica dataset. Fig. 7 visualizes the queried points for a scene in the Replica dataset given the query "cloth". We observe that LangSplat-m and GaussianGrouping-m failed to retrieve the correct object, and OpenGaussian only retrieves part of the cloth with noisy points from other objects. Overall, our method retrieved a more complete and clearer object, which is consistent with the quantitative result.

C. Ablation study.

We conduct ablation studies to validate the efficacy of each proposed component of our method and report the performance in Tab. IV on LERF's "figurines" scene. Recall, $\bar{\phi}_r$ denotes the *weighted* average features. As a comparison, we also tested on a standard average strategy that is the sum of features from various views divided by the number of features, denoted as $\bar{\phi}_r^{\texttt{stand.}}$. As shown, mIoU increases significantly, *i.e.* +49.89, when using the GT-anchored query. In the meantime, SAM2's masklets to get consistent features of the region also play an important role for the GT-anchored query to work.

Overall, we observe that all proposed components are necessary to achieve the optimal performance. We also studied the effectiveness of our method without DBSCAN [5] and evaluated the performance of canonical querying from LERF [11] on the task of 3D querying. The original definition of canonical query from LERF is conducted on rendered 2D images.

To evaluate the performance of canonical query in 3D querying, we replace $\phi_{\texttt{lang}}$ by $\boldsymbol{l}_i$ in the original equation. Formally, for each language embedding $\boldsymbol{l}_i$ and each text query $\boldsymbol{q}$, the relevancy score is defined as $\min_i \frac{\exp(\boldsymbol{l}_i \cdot \boldsymbol{q})}{\exp(\boldsymbol{l}_i \cdot \boldsymbol{q}) + \exp(\boldsymbol{q} \cdot \phi^i_{canon})}$ where ϕ^i_{canon} is the CLIP embedding of a predefined *canonical* phrase chosen from *"object"*, *"things"*, *"stuff"*, and *"texture"*. The results are shown in Tab. V on LERF's "figurines" scene. We see that without DBSCAN, our proposed method also gains significant improvement in performance. We also see that our proposed GT-anchored query significantly outperforms the canonical query.

VI. Conclusion

We study the task of open-vocabulary 3D understanding formulated as a point-level 3D querying task. Based on LangSplat's framework, we present a tracking-based approach to provide consistent ground-truth supervision when distilling the language features. Furthermore, we introduce a novel GT-Anchored querying pipeline to address the difficulty of choosing a consistent threshold in the baseline's single-step query. Experiments over three datasets demonstrate that our approach achieves state-of-the-art performance, with ablation studies verifying the efficacy of the proposed components.

References

[1] Mathilde Caron, Hugo Touvron, Ishan Misra, Hervé Jégou, Julien Mairal, Piotr Bojanowski, and Armand Joulin. Emerging properties in self-supervised vision transformers. In *ICCV*, 2021. 2

[2] Jiazhong Cen, Zanwei Zhou, Jiemin Fang, Wei Shen, Lingxi Xie, Dongsheng Jiang, Xiaopeng Zhang, Qi Tian, et al. Segment anything in 3d with NeRFs. In *NeurIPS*, 2023. 2

[3] Guikun Chen and Wenguan Wang. A survey on 3D Gaussian splatting. *arXiv preprint arXiv:2401.03890*, 2024. 1

[4] Mehdi Cherti, Romain Beaumont, Ross Wightman, Mitchell Wortsman, Gabriel Ilharco, Cade Gordon, Christoph Schuhmann, Ludwig Schmidt, and Jenia Jitsev. Reproducible scaling laws for contrastive language-image learning. In *CVPR*, 2023. 3

[5] Martin Ester, Hans-Peter Kriegel, Jörg Sander, Xiaowei Xu, et al. A density-based algorithm for discovering clusters in large spatial databases with noise. In *KDD*, 1996. 3, 7

[6] Jun Guo, Xiaojian Ma, Yue Fan, Huaping Liu, and Qing Li. Semantic Gaussians: Open-vocabulary scene understanding with 3D Gaussian splatting. *arXiv preprint arXiv:2403.15624*, 2024. 2

[7] Shuting He, Henghui Ding, Xudong Jiang, and Bihan Wen. SegPoint: Segment any point cloud via large language model. In *ECCV*, 2024. 2

[8] Krishna Murthy Jatavallabhula, Alihusein Kuwajerwala, Qiao Gu, Mohd Omama, Tao Chen, Alaa Maalouf, Shuang Li, Ganesh Iyer, Soroush Saryazdi, Nikhil Keetha, et al. ConceptFusion: Open-set multimodal 3D mapping. *RSS*, 2023. 1, 2

[9] Shengxiang Ji, Guanjun Wu, Jiemin Fang, Jiazhong Cen, Taoran Yi, Wenyu Liu, Qi Tian, and Xinggang Wang. Segment any 4d Gaussians. *arXiv preprint arXiv:2407.04504*, 2024. 2

[10] Bernhard Kerbl, Georgios Kopanas, Thomas Leimkühler, and George Drettakis. 3D Gaussian splatting for real-time radiance field rendering. *ACM TOG*, 2023. 1, 2

[11] Justin Kerr, Chung Min Kim, Ken Goldberg, Angjoo Kanazawa, and Matthew Tancik. LERF: Language embedded radiance fields. In *ICCV*, 2023. 1, 2, 5, 7

[12] Chung Min Kim, Mingxuan Wu, Justin Kerr, Ken Goldberg, Matthew Tancik, and Angjoo Kanazawa. GARField: Group anything with radiance fields. In *CVPR*, 2024. 2

[13] 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 *ICCV*, 2023. 2, 3

[14] Will LeVine, Benjamin Pikus, Pranav Raja, and Fernando Amat Gil. Enabling calibration in the zero-shot inference of large vision-language models. In *ICLR Workshop on Trustworthy ML*, 2023. 5

[15] Boyi Li, Kilian Q Weinberger, Serge Belongie, Vladlen Koltun, and René Ranftl. Language-driven semantic segmentation. *ICLR*, 2022. 1, 2

[16] 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

[17] Kunhao Liu, Fangneng Zhan, Jiahui Zhang, Muyu Xu, Yingchen Yu, Abdulmotaleb El Saddik, Christian Theobalt, Eric Xing, and Shijian Lu. Weakly supervised 3d open-vocabulary segmentation. In *NeurIPS*, 2023. 1, 2, 5

[18] Youquan Liu, Lingdong Kong, Jun Cen, Runnan Chen, Wenwei Zhang, Liang Pan, Kai Chen, and Ziwei Liu. Segment any point cloud sequences by distilling vision foundation models. In *NeurIPS*, 2024. 2

[19] Ricardo Martin-Brualla, Noha Radwan, Mehdi SM Sajjadi, Jonathan T Barron, Alexey Dosovitskiy, and Daniel Duckworth. Nerf in the wild: Neural radiance fields for unconstrained photo collections. In *CVPR*, 2021. 1

[20] 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. *CACM*, 2021. 1, 2

[21] Minghan Qin, Wanhua Li, Jiawei Zhou, Haoqian Wang, and Hanspeter Pfister. LangSplat: 3d language Gaussian splatting. In *CVPR*, 2024. 1, 2, 3, 4, 5, 6

[22] 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 *ICML*, 2021. 1, 2

[23] Nikhila Ravi, Valentin Gabeur, Yuan-Ting Hu, Ronghang Hu, Chaitanya Ryali, Tengyu Ma, Haitham Khedr, Roman Rädle, Chloe Rolland, Laura Gustafson, et al. SAM 2: Segment anything in images and videos. In *ICLR*, 2025. 2, 4

[24] William Shen, Ge Yang, Alan Yu, Jansen Wong, Leslie Pack Kaelbling, and Phillip Isola. Distilled feature fields enable few-shot language-guided manipulation. In *CORL*, 2023. 2

[25] Julian Straub, Thomas Whelan, Lingni Ma, Yufan Chen, Erik Wijmans, Simon Green, Jakob J Engel, Raul Mur-Artal, Carl Ren, Shobhit Verma, et al. The replica dataset: A digital replica of indoor spaces. *arXiv preprint arXiv:1906.05797*, 2019. 1, 5

[26] 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. *NeurIPS*, 2024. 1, 2, 3, 5, 6

[27] Yiheng Xie, Towaki Takikawa, et al. Neural fields in visual computing and beyond. In *Computer Graphics Forum*, 2022. 1

[28] Xiuwei Xu, Huangxing Chen, Linqing Zhao, Ziwei Wang, Jie Zhou, and Jiwen Lu. Embodiedsam: Online segment any 3d thing in real time. *ICLR*, 2024. 1, 2

[29] Kashu Yamazaki, Taisei Hanyu, Khoa Vo, Thang Pham, Minh Tran, Gianfranco Doretto, Anh Nguyen, and Ngan Le. Open-fusion: Real-time open-vocabulary 3d mapping and queryable scene representation. In *ICRA*, 2024. 2

[30] Jianglong Ye, Naiyan Wang, and Xiaolong Wang. Featurenerf: Learning generalizable nerfs by distilling foundation models. In *ICCV*, 2023. 2

[31] Mingqiao Ye, Martin Danelljan, Fisher Yu, and Lei Ke. Gaussian grouping: Segment and edit anything in 3d scenes. In *ECCV*, 2024. 1, 2, 5, 6

[32] Zehao Yu, Anpei Chen, Binbin Huang, Torsten Sattler, and Andreas Geiger. Mip-Splatting: Alias-free 3D Gaussian splatting. In *CVPR*, 2024. 1

[33] Shuaifeng Zhi, Tristan Laidlow, Stefan Leutenegger, and Andrew J Davison. In-place scene labelling and understanding with implicit scene representation. In *ICCV*, 2021. 2

[34] Chong Zhou, Chen Change Loy, and Bo Dai. Extract free dense labels from CLIP. In *ECCV*, 2022. 2

[35] Shijie Zhou, Haoran Chang, et al. Feature 3DGS: Supercharging 3d Gaussian splatting to enable distilled feature fields. In *CVPR*, 2024. 1, 2