Visibility-Aware Language Aggregation for Open-Vocabulary Segmentation in 3D Gaussian Splatting

Sen Wang[1,2] Kunyi Li[1,2] Siyun Liang[1,5] Elena Alegret [1] Jing Ma [4]
Nassir Navab[1,2] Stefano Gasperini[1,2,3]
[1]Technical University of Munich [2]Munich Cental for Machine Learning [3]VisualAIs
[4]Ludwig Maximilian University of Munich [5]University of Tübingen

Abstract

Recently, distilling open-vocabulary language features from 2D images into 3D Gaussians has attracted significant attention. Although existing methods achieve impressive language-based interactions with 3D scenes, we observe two fundamental issues: background Gaussians, which contribute negligibly to a rendered pixel, receive the same feature as the dominant foreground ones, and multi-view inconsistencies due to view-specific noise in language embeddings. We introduce Visibility-Aware Language Aggregation (VALA), a lightweight yet effective method that computes marginal contributions for each ray and applies a visibility-aware gate to retain only visible Gaussians. Moreover, we propose a streaming weighted geometric median in cosine space to merge noisy multi-view features. Our method yields a robust, view-consistent language feature embedding in a fast and memory-efficient manner. VALA improves open-vocabulary localization and segmentation across reference datasets, consistently surpassing existing works. The source code is available on VALA.

1. Introduction

Understanding 3D scenes is essential for interacting with the environment in robotic navigation [2, 23], autonomous driving [8, 32], and augmented reality [10, 17]. Traditional approaches, however, are constrained to a fixed set of object categories defined at training time [4, 27, 35], limiting their applicability to open-world scenarios. Thanks to recent advances in vision-language models [11, 30], open-vocabulary methods [9, 24, 42] enable querying and interacting with 3D scenes through natural language, and recognizing unseen object categories without requiring retraining.

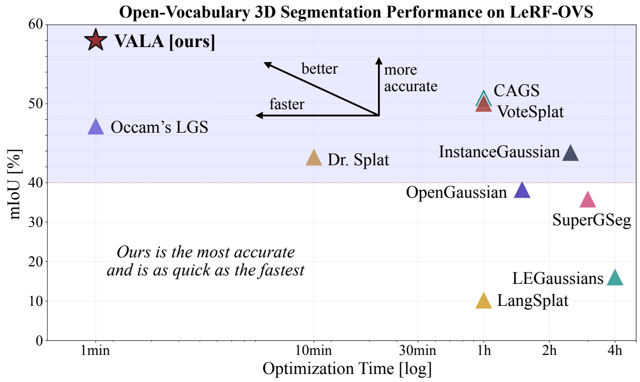


Figure 1. Thanks to its feature aggregation that is visibility-aware and multi-view consistent, our proposed VALA is the most accurate and as quick as the fastest [3] to optimize. Comparison in 3D open-vocabulary segmentation on the LeRF-OVS dataset [28].

While classical 3D understanding methods operate on point clouds or meshes derived from 3D sensors, recent neural scene representations, such as NeRFs [22] and 3D Gaussian Splatting (3DGS) [14], have emerged as a compelling alternative. They not only enable high-quality rendering from novel viewpoints but also facilitate semantic reasoning, as appearance and geometry are encoded jointly. Thus, open-vocabulary reasoning has recently been grounded in neural 3D scene representations [15, 28], enabling new semantic interactions in 3D environments. Initially explored with NeRFs [7, 15], the efficiency and explicit nature of 3DGS simplified the integration of semantic features, contributing to its widespread adoption [3, 13, 28, 38].

At the core of these approaches lies the challenge of embedding reliable semantic and language features into the 3D representation. Current methods rely on powerful off-the-shelf 2D foundation models, such as SAM [16] and CLIP [30], which produce 2D feature maps that must be lifted to 3D and aggregated across views. Proper aggregation is critical for accurate 3D segmentation.

Despite numerous recent advances [12, 13, 18, 34], current approaches suffer from an inherent limitation: they assign 2D features indiscriminately to *all* Gaussians along a camera ray, disregarding scene geometry and occlusion relationships. Consequently, features originating from foreground objects (*e.g.*, a vase) are incorrectly propagated to background structures (*e.g.*, the supporting table or floor), leading to substantial degradation in open-vocabulary recognition accuracy.

Furthermore, when lifted into 3D, 2D features exhibit multi-view inconsistencies. The same object may produce divergent feature representations across different viewpoints, a phenomenon known as semantic drift [15]. Current methods address this by promoting cross-view consistency through 3D-consistent clustering and contrastive objectives derived from SAM masks [18, 20, 26, 38]. Nevertheless, such strategies generally require extensive per-scene optimization, and their heavy reliance on noisy, view-dependent 2D cues often undermines cluster reliability.

In this paper, we address these fundamental feature aggregation problems with VALA (Visibility-Aware Language Aggregation), a lightweight yet effective framework that combines a two-stage gating mechanism with a robust multi-view feature aggregation strategy. Our gating mechanism leverages the statistical distribution of per-ray Gaussian contributions (termed visibility) to preferentially propagate features to Gaussians with high visibility, thereby ensuring accurate feature assignment. To further mitigate multi-view inconsistencies in 2D language features, we introduce a convex but non-smooth optimization on the unit hypersphere, which we reformulate into a streaming gradient-based procedure that achieves consistent embeddings without additional computational overhead. As shown in Figure 1, VALA strategies are highly effective.

Our contributions can be summarized as follows:

- We identify fundamental issues in the feature aggregation of current works as a bottleneck in open-vocabulary 3D scene understanding.
- We introduce VALA, a visibility-aware feature propagation framework that employs a two-stage gating mechanism to assign features based on Gaussian visibility.
- We propose a robust aggregation strategy for the 2D features using the streaming cosine median, thereby improving multi-view consistency.
- We obtain state-of-the-art performance in 2D *and* 3D on open-vocabulary segmentation for 3DGS scenes on the reference datasets LeRF-OVS [28] and ScanNet-v2 [5].

2. Related works

Open-Vocabulary Feature Distillation. Recent works have embedded 2D vision-language features into 3D scene representations to enable open-vocabulary 3D understanding. Pioneering efforts on NeRFs such as LERF [15] and OpenNeRF [7] used CLIP [30] embeddings and pixel-aligned features, enabling open-vocabulary queries. However, due to the computational needs of NeRF [22], they face scalability and efficiency bottlenecks. Thus, subsequent works have embedded language features into 3DGS [31, 41, 43]. LangSplat [28] employs SAM [16] to extract multi-level CLIP features, then compresses dimensionality with an autoencoder to build a compact yet expressive 3D language field. Feature3DGS [41] uses a convolutional neural network (CNN) to lift feature dimensions. Although both approaches aim to compress the supervision signal, this dimensionality reduction inevitably results in information loss. GOI [29] and CCL-LGS [36] employ a single trainable feature codebook to store language embeddings, with an MLP predicting discrete codebook indices for rasterized 2D feature maps, which compress semantics spatially rather than dimensionally and retain semantic richness. However, as these approaches rely on 2D rendered feature maps for perception, their performance in 3D scene understanding is significantly limited.

Other methods first group 3D Gaussians or points into semantically meaningful clusters, typically corresponding to objects or parts, and then assign a language feature to each cluster as a whole [12, 18, 20, 26, 29, 38]. These methods introduce an explicit discrete grouping step as a form of prior semantic structuring: OpenGaussian [38] performs coarse-to-fine clustering based on spatial proximity followed by feature similarity. SuperGSeg [20] and InstanceGaussian [18] both leverage neural Gaussians to model instance-level features: SuperGSeg groups Gaussians into Super-Gaussians to facilitate language assignment, whereas InstanceGaussian directly assigns fused semantic features to each cluster. VoteSplat [12] and OpenSplat3D [26] mitigate the pixel-level ambiguities of the direct distillation. Then, the resulting cluster graph structures support higher-level reasoning [20, 40], which per-Gaussian features cannot easily enable. However, all these methods rely on feature distillation using per-cluster learnable language embeddings. These approaches are computationally expensive and highly sensitive to noise or outliers in the preprocessed feature maps, since the language features are optimized directly in Euclidean space. As a result, even minor errors in the input features can propagate through the model, leading to inconsistent or inaccurate semantic representations, particularly in complex or cluttered scenes.

Open-Vocabulary Feature Aggregation. Beyond cluster-based language features distillation, recent works adopt more efficient strategies for feature aggregation. For instance, Dr.Splat [13] and Occam's LGS [3] bypass intermediate 2D supervision and clustering by directly injecting language features into 3D Gaussians, achieving fast, accurate results in a training-free regime. While these direct feature aggregation methods deliver strong runtime efficiency and segmentation accuracy, they indiscriminately propagate 2D features to *every* Gaussian intersected by each camera ray, disregarding scene geometry and occlusion. As a result, features from foreground objects (e.g., a vase) are erroneously assigned to background elements (e.g., the table

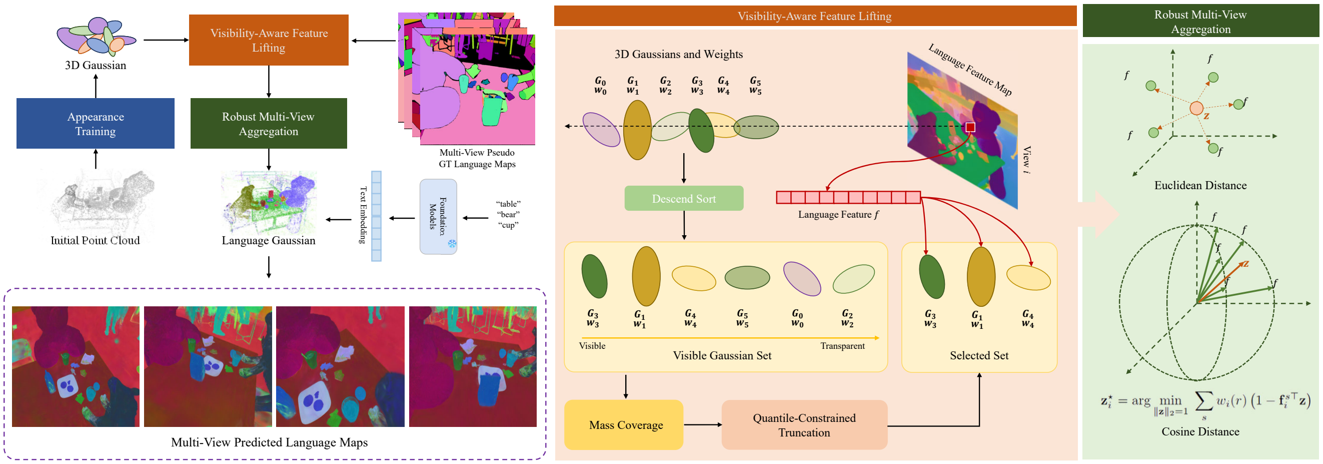


Figure 2. Overview of VALA. The framework is shown on the left, with the orange and green blocks detailed on the right being our key contributions: the visibility-aware feature lifting (orange, Section 4.1), and the robust multi-view aggregation (green, Section 4.2).

or floor). Moreover, existing methods share two critical limitations: (i) they assign equal supervision to all Gaussians along a ray, ignoring each Gaussian's marginal contribution to the rendered pixel, and (ii) they overlook the view-dependent noise and inconsistency in 2D language features. We address these issues with VALA, a robust and efficient training-free framework that improves open-vocabulary grounding through visibility-aware gating (for contribution-aligned supervision) and robust multi-view aggregation.

3. Preliminaries

We briefly recall 3DGS [14] and how the features are assigned to a 3D Gaussian without iterative training.

3D Gaussian Primitives and Projection. A scene is represented by a set of anisotropic Gaussians $\mathcal{G} = \{g_i\}_{i=1}^N$, with each Gaussian featured with $g_i = (\boldsymbol{\mu}_i, \boldsymbol{\Sigma}_i, \mathbf{c}_i, o_i)$, where $\boldsymbol{\mu}_i \in \mathbb{R}^3$ and $\boldsymbol{\Sigma}_i \in \mathbb{R}^{3\times3}$ are the mean position and covariance matrix $\mathbf{c}_i$ encodes appearance (*e.g.*, RGB or spherical harmonics coefficients), and $o_i \in (0, 1]$ is a base opacity.

Images are rasterized by splatting the Gaussians from near to far along the camera ray through pixel u, followed by front-to-back α-blending the Gaussian contributions, as:

$$\alpha_i(\mathbf{u}) = 1 - \exp\big(o_i \rho_i(\mathbf{u})\big), \tag{1}$$

$$T_i(\mathbf{u}) = \prod\nolimits_{j<i} \big(1 - \alpha_j(\mathbf{u})\big), \tag{2}$$

$$\mathbf{C}(\mathbf{u}) = \sum\nolimits_i \alpha_i(\mathbf{u})\, T_i(\mathbf{u})\, \mathbf{c}_i(\mathbf{u}), \tag{3}$$

where $\rho_i(\mathbf{u})$ is the projected 2D Gaussian density in screen space, with projected 2D mean $\tilde{\boldsymbol{\mu}}_i$ and covariance $\tilde{\boldsymbol{\Sigma}}_i$, and

$$\rho_i(\mathbf{u}) = \exp\Big(-\tfrac{1}{2}(\mathbf{u} - \tilde{\boldsymbol{\mu}}_i)^\top \tilde{\boldsymbol{\Sigma}}_i^{-1} (\mathbf{u} - \tilde{\boldsymbol{\mu}}_i)\Big). \tag{4}$$

We denote the *marginal contribution* of g_i to pixel $\mathbf{u}$ as

$$w_i(\mathbf{u}) \;=\; \alpha_i(\mathbf{u})\, T_i(\mathbf{u}). \tag{5}$$

Language Features Assignment via Direct Aggregation. Recent works [3, 13] proposed to directly assign 2D language features to 3D Gaussians via weighted feature aggregation. To obtain training-free 3D language feature embeddings, Kim *et al.* [13] pool per-pixel weights $w_i(I, r)$, defined as in Eq. (5), using segmentation masks $M_j(I, r)$:

$$w_{ij} = \sum\nolimits_{I\in\mathcal{I}} \sum\nolimits_{r\in\Omega_I} M_j(I, r) \cdot w_i(I, r), \tag{6}$$

where w_{ij} associates Gaussian i-mask j, and Ω_I is the pixel domain of image I. The final CLIP embedding for each i is a weighted average over the mask-level embeddings f_j^{map}:

$$f_i = \sum\nolimits_{j=1}^{M} \frac{w_{ij}}{\sum_{k=1}^{M} w_{ik}} f_j^{\mathrm{map}}. \tag{7}$$

Although this mask-based aggregation is a straightforward way to lift CLIP features into 3D, it has a memory footprint that scales quadratically with the scene complexity. To overcome this limitation, we adopt Occam's LGS [3]'s probabilistic per-view aggregation strategy as our baseline. [3] avoids explicit mask representations and dense weight storage, maintaining semantic consistency across views. So, the 3D feature f_i for Gaussian i becomes:

$$f_i = \frac{\sum_{s\in\mathcal{S}_i} w_i^s f_i^s}{\sum_{s\in\mathcal{S}_i} w_i^s}, \tag{8}$$

where $\mathcal{S}_i$ is the views set where Gaussian i is visible, w_i^s is the marginal contribution of i at its center projection in view s, and f_i^s is the 2D feature at the corresponding pixel.

4. Method

We aim to distill language features into 3DGS under *visibility constraints*, to get semantically rich and *view-consistent* 3D embeddings. Existing approaches that indiscriminately assign identical 2D features to *all* Gaussians along a camera ray, which leads to noisy supervision and cross-view inconsistencies. With VALA, we assign only visible features.

Our pipeline is shown in Figure 2. Built on a direct feature assignment method, VALA has two complementary components to improve the assignment of 2D vision-language features to the 3D scene. First, we introduce a *visibility-aware attribution* mechanism to selectively assign language features to Gaussians based on their relevance in the rendered scene (Section 4.1). Second, we propose a *robust cross-view consolidation* strategy to aggregate per-view features while suppressing inconsistent observations, yielding coherent 3D semantic embeddings (Section 4.2).

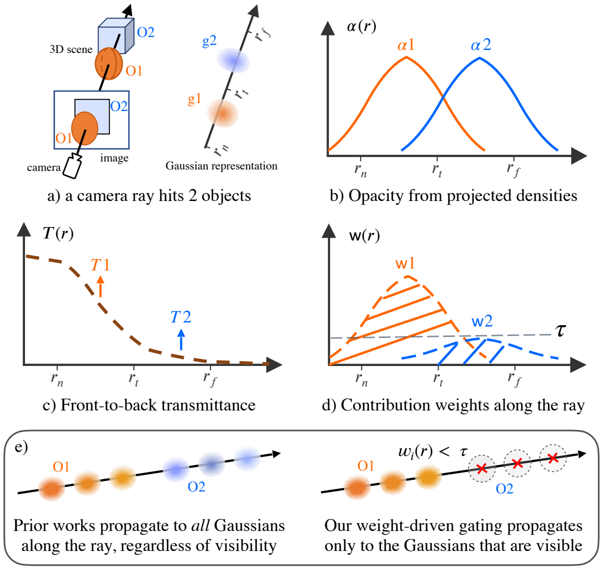
4.1. Visibility-Aware Feature Lifting

Recent works explored lifting 2D language embeddings into 3D space via differentiable rendering pipelines [3, 13]. However, existing approaches assign the same 2D language feature to *all* Gaussians intersected by a given pixel ray, regardless of each Gaussian's actual contribution to the rendered pixel. As illustrated in Figure 3, when an object O_2 is occluded by another object O_1, the 2D language embedding at that pixel primarily represents the semantics of O_1. Nevertheless, a Gaussian g_2 belonging to O_2 may still be incorrectly associated with the language feature of O_1.

This erroneous assignment occurs in both alpha-blending-based language assignment methods [20, 28] and, more prominently, in direct feature assignment methods [3, 13, 38]. As shown in Figure 3 (b–c), even though the transmittance (Eq. (2)) decreases monotonically along the ray from near to far—resulting in a very small transmittance for g_2—its alpha value (Eq. (1)) can remain relatively large in the far region. This, yields a non-negligible compositing weight (Eq. (9)) for g_2, which, according to Eq. (7) or Eq. (8), contributes substantially to the final aggregated feature of g_2. Such unintended contributions introduce ambiguity into the 3D representation.

Recent works have introduced changes that indirectly affect this assignment. Dr.Splat [13] selects the top-k Gaussians along each ray, but this reduces computational costs rather than ensuring the correct semantic allocation. VoteSplat [12] recognizes that distant Gaussians may suffer from occlusion, but discards the compositing weights altogether and instead averages the features of *all* intersected Gaussians to generate 3D votes for the clustering step. While they may tangentially bring improvements, they leave unsolved the assignment problem described above and continue to *propagate wrong features* to background regions.

To overcome this limitation, we introduce a visibility-aware gating mechanism, which selectively supervises only the Gaussians along each ray that contribute to the pixel. By leveraging per-ray visibility weights, our method filters out occluded or low-contribution Gaussians before aggregating the features, ensuring that only geometrically and photometrically relevant points receive semantic supervision. First, we clarify how we compute the *per-ray weights*.



a) a camera ray hits 2 objects

b) Opacity from projected densities

c) Front-to-back transmittance

d) Contribution weights along the ray

Prior works propagate to *all* Gaussians along the ray, regardless of visibility

Our weight-driven gating propagates only to the Gaussians that are visible

Figure 3. **Visibility-aware gating for semantic assignment** (Section 4.1). Simplified representation of a scene with two objects (a) O_1, O_2 and a camera ray r with Gaussians g_1, g_2. We compute the opacity (b) and compute the transmittance front-to-back (c). Then we calculate the contribution weights for each ray, thresholding with τ (d). Instead of propagating the features to *all* Gaussians as prior works do, our gating only propagates to the visible ones (e).

Ray Notation and Marginal Contributions. Let r denote the camera ray through pixel $\mathbf{u}$. For brevity, we write

$$\begin{aligned} &T_i(r) \equiv T_i(\mathbf{u}), && \alpha_i(r) \equiv \alpha_i(\mathbf{u}), \\ &w_i(r) \equiv \alpha_i(r)\, T_i(r). && \end{aligned} \tag{9}$$

where $\alpha_i(r)$ encodes *coverage* (*i.e.*, how much g_i overlaps the pixel), $T_i(r)$ represents *transmittance* (*i.e.*, how much light reaches g_i after occlusion by nearer Gaussians), and $w_i(r)$ measures how strongly g_i influences the rendered sample along r. We name this as the *Visibility* of a Gaussian from a specific view. Instead of assigning this feature to all Gaussians on the ray r, we use a two-stage visibility-aware gate (VAG). We aggregate the weights into a per-view visibility score

$$S^s_{\text{tot}} \;=\; \sum\nolimits_{i,r} w_i(r). \tag{10}$$

Stage A: Mass Coverage on the Thresholded Set. We sort $\{w_i(r)\}_i$ decreasingly, with the indices as $(1), \ldots, (k)$. We then retain the shortest prefix that accounts for a target fraction $\tau_{\text{view}} \in [0.5, 0.75]$ of the total visibility mass:

$$k^\star_{\text{mass}} \;=\; \min\Big\{k : \sum\nolimits_{j=1}^{k} w_j \;\geq\; \tau_{\text{view}}\; S^s_{\text{tot}}\Big\}. \tag{11}$$

To suppress numerical noise, we apply a small absolute floor τ_{abs} and define the candidate set as

$$\mathcal{G}^s_{\text{mass}} = \Big\{(1), \ldots, (k^\star_{\text{mass}})\Big\} \;\cap\; \{i :\; w_i \geq \tau_{\text{abs}}\}. \tag{12}$$

Stage B: Quantile-Constrained Truncation. Let $\tau_q^s = \mathrm{Quantile}_{1-q}(\{w_i\}_i)$, we define $K_q^s = |\{i:\ w_i \geq \tau_q^s\}|$ and instead of imposing a separate hard limit, we determine the selection cap directly via the q-quantile as

$$\begin{aligned} k^\star_{\text{keep}} &= \min\big(k^\star_{\text{mass}},\ K_q^s\big),\\ \mathcal{G}^s_{\text{keep}} &= \big\{(1),\dots,(k^\star_{\text{keep}})\big\}. \end{aligned} \tag{13}$$

Why Mass *then* Quantile? A fixed quantile alone tightly controls cardinality but ignores how visibility mass is distributed, and under heavy tails may discard essential contributors. Conversely, mass coverage secures a target fraction of visible content but can be liberal when scores are flat. Our two-stage rule reconciles both: Stage A guarantees coverage on the *relevant* (floored) set, while B imposes a quantile-derived *cardinality constraint* K_q^s that stabilizes scale across views. Practically, if $K_q^s \geq k^\star_{\text{mass}}$, we keep the mass-coverage set unchanged; otherwise we truncate it to the top-K_q^s by w_i. The gate is thus *coverage-faithful* and *scale-adaptive*.

4.2. Robust Multi-View Aggregation

SAM+CLIP preprocessing pipelines [28] yield crisp mask boundaries and per-pixel open-vocabulary embeddings, but their semantics are often viewpoint-dependent: changes in viewpoint and occlusion induce noticeable drift across views. To enforce multi-view consistency, several 3DGS-based methods first form 3D-consistent clusters, typically supervised with contrastive signals derived from SAM masks, and then assign a language embedding to each cluster [18, 20, 26, 38]. While this decoupled clustering can improve multi-view semantic consistency, it makes the pipelines' training multi-stage, thus prolonging the training time. More critically, because clustering is still driven by noisy, view-dependent 2D cues, it does not correct the root cause, namely, upstream semantic drift, which can bias the clusters and ultimately degrade the accuracy of the final language assignments.

To address this multi-view inconsistency at source, we adopt geometric median [1, 21, 37] to robustly aggregate multi-view features by minimizing the cosine distances in feature space. Unlike aggregation by weighted mean, it dampens view-dependent outliers and semantic drift.

Weighted Euclidean Geometric Median. Using the visibility weights defined in Eq. (9), the (weighted) geometric median for g_i is

$$\mathbf{z}_i^\star = \mathrm{argmin}_{\mathbf{z}\in\mathbb{R}^d} \sum\nolimits_s w_i(r)\,\big\|\mathbf{z}-\mathbf{f}_i^s\big\|. \tag{14}$$

Cosine-loss Median on the Unit Sphere. $\mathbf{f}(I,\mathbf{u})$ are ℓ_2-normalized embeddings and thus angular consistency is most relevant. Therefore, we constrain $\mathbf{z}_i$ to the unit sphere $\mathbb{S}^{d-1}$ and minimize a weighted cosine loss:

$$\mathbf{z}_i^\star = \mathrm{argmin}_{\|\mathbf{z}\|_2=1} \sum\nolimits_s w_i(r)\,\big(1-\mathbf{f}_i^{s\top}\mathbf{z}\big), \tag{15}$$

where $w_i(r)$ denotes the visibility weight of Gaussian g_i from view s, since r represents the view s. The gradient of $\ell(\mathbf{f},\mathbf{z}) = 1-\mathbf{f}^\top\mathbf{z}$ on $\mathbb{S}^{d-1}$ is $\nabla_{\mathbf{z}}\ell = -\big[\mathbf{f}-(\mathbf{f}^\top\mathbf{z})\,\mathbf{z}\big]$, the projection of $\mathbf{f}$ onto the tangent space at $\mathbf{z}$. Compared to the Euclidean formulation in Eq. (14), this objective directly optimizes angular similarity, circumventing the scale sensitivity of Euclidean distances in high dimensions, where norm variations dominate over angular differences, and empirically leads to more stable 3D semantics (Table 3).

Algorithm 1 Streaming cosine-loss median on $\mathbb{S}^{d-1}$ (Section 4.2).

```
Require: Stream $\{(\mathbf{f}_t, w_i^t)\}_{t=1}^T$ with $\mathbf{f}_t \in \mathbb{R}^d$, $\|\mathbf{f}_t\|_2 = 1$,
    and $w_i^t > 0$
Initialize $\mathbf{z}_{i,0} \leftarrow \mathbf{f}_1, \quad W_{i,0} \leftarrow 0$
for $t = 1, \dots, T$ do
    $\mathbf{d}_t \leftarrow \mathbf{f}_t - (\mathbf{f}_t^\top \mathbf{z}_{i,t})\,\mathbf{z}_{i,t}$        ▷ tangent direction
    $\eta_t \leftarrow \dfrac{w_i^t}{W_{i,t} + w_i^t}$        ▷ streaming step size
    $\mathbf{z}_{i,t+1} \leftarrow \mathrm{Norm}\big(\mathbf{z}_{i,t} + \eta_t\,\mathbf{d}_t\big)$
    $W_{i,t+1} \leftarrow W_{i,t} + w_i^t$
end for
return $\mathbf{z}_i \leftarrow \mathbf{z}_{i,T},\ W_i \leftarrow W_{i,T}$
```

Constant-Memory Streaming Update. While effective, solving Eq. (15) with the classical Weiszfeld algorithm [6] requires repeated full-batch updates over all Gaussian features, which scales linearly with the number of views and becomes computationally prohibitive in practice. To address this, we adopt a constant-memory streaming scheme inspired by online optimization [16]. Specifically, as detailed in Algorithm 1, we maintain only the current estimate $(\mathbf{z}_{i,t}, W_{i,t})$, where $W_{i,t}$ is the cumulative visibility weight, and incorporate each new observation $(\mathbf{f}_t, w_i^t)$ via

$$\mathbf{z}_{i,t+1} = \mathrm{Norm}\big(\mathbf{z}_{i,t} + \eta_t\, w_i^t\,\big[\mathbf{f}_t - (\mathbf{f}_t^\top \mathbf{z}_{i,t})\,\mathbf{z}_{i,t}\big]\big)\,, \tag{16}$$

$$\eta_t = \frac{w_i^t}{W_{i,t}+w_i^t}\,, \qquad W_{i,t+1} = W_{i,t} + w_i^t, \tag{17}$$

where $\mathrm{Norm}(\mathbf{x}) = \mathbf{x}/\|\mathbf{x}\|_2$ projects $\mathbf{z}_{i,t}$ onto the unit sphere $\mathbb{S}^{d-1}$. The update direction $\mathbf{f}_t - (\mathbf{f}_t^\top \mathbf{z}_{i,t})\mathbf{z}_{i,t}$ lies in the tangent space and increases cosine similarity, while the adaptive step size η_t weights each sample according to its visibility. Under standard stochastic approximation assumptions, $\mathbf{z}_{i,t}$ converges to a stationary point of Eq. (15) at rate $\mathcal{O}(1/\sqrt{W_{i,t}})$.

5. Experiments

5.1. Experimental setup

Datasets. We evaluate on the two reference datasets for this task: LERF-OVS [28] and ScanNet-v2 [5]. LERF-OVS is derived from the LERF dataset of Kerr *et al.* [15],

		Mean		Figurines		Ramen		Teatime		Waldo_Kitchen	
	Method	mIoU	mAcc	mIoU	mAcc	mIoU	mAcc	mIoU	mAcc	mIoU	mAcc
2D evaluation	LERF [15]	37.4	73.6	38.6	75.0	28.2	62.0	45.0	84.8	37.9	72.7
	LEGaussian [31]	24.6	67.4	23.4	57.1	20.2	69.0	32.3	79.7	22.3	63.6
	GOI [29]	42.0	59.2	23.9	44.6	33.7	56.3	55.8	67.8	54.5	68.2
	GAGS [25]	54.1	81.7	53.6	78.6	46.8	69.0	60.3	88.1	55.8	90.9
	LangSplat [28]	51.4	84.3	44.7	80.4	51.2	73.2	65.1	88.1	44.5	95.5
	LangSplatV2 [19]	59.9	84.1	56.4	82.1	51.8	74.7	72.2	93.2	59.1	86.4
	Occam's LGS [3]	61.3	82.5	58.6	80.4	51.0	74.7	70.2	93.2	65.3	81.8
	VALA [ours]	61.7	86.4	59.9	82.1	51.5	75.6	70.2	91.5	65.1	86.4
3D evaluation	LangSplat [28]	10.35	13.64	7.27	10.71	10.05	9.86	14.38	20.34	9.71	9.09
	LEGaussians [31]	16.21	23.82	17.99	23.21	15.79	26.76	19.27	27.12	11.78	18.18
	OpenGaussian [38]	38.36	51.43	39.29	55.36	31.01	42.25	60.44	76.27	22.70	31.82
	SuperGSeg [20]	35.94	52.02	43.68	60.71	18.07	23.94	55.31	77.97	26.71	45.45
	Dr.Splat [13]	43.29	64.30	54.42	80.36	24.33	35.21	57.35	77.97	37.05	63.64
	InstanceGaussian [18]	43.87	61.09	54.87	73.21	25.03	38.03	54.13	69.49	41.47	63.64
	CAGS [34]	50.79	69.62	60.85	82.14	36.29	46.48	68.40	86.44	37.62	63.64
	VoteSplat [12]	50.10	67.38	68.62	85.71	39.24	61.97	66.71	88.14	25.84	33.68
	Occam's LGS [3]	47.22	74.84	52.90	78.57	32.01	54.92	61.02	93.22	42.95	72.72
	VALA [ours]	58.02	82.85	60.38	89.29	45.41	67.61	70.61	88.14	55.71	86.36

Table 1. Comparison on LERF-OVS (mIoU / mAcc.). In 3D, results are taken from [12, 13, 20, 34, 38] and otherwise evaluated by us.

where we evaluate open-vocabulary object selection in both 2D and 3D. For the 2D evaluation, we follow the protocol of LERF [15]. For the 3D evaluation, we follow OpenGaussian [38]. On ScanNet, we evaluate 3D semantic segmentation. Previous evaluation protocols [18, 38] freeze the growth of 3D Gaussians, which degrades photometric fidelity. In contrast, we allow full optimization of the 3D Gaussians, resulting in misalignment between the optimized Gaussians and the ground-truth point cloud. We therefore adapt the evaluation protocol in [13] by propagating pseudo ground-truth labels to the Gaussians. Details are provided in the Appendix B.

Implementation Details. We generate SAM [16] masks at subpart, part, and whole object granularities. We use OpenCLIP ViT-B/16 [30] and the `gsplat` rasterizer [39]. We apply direct feature aggregation in the 512-dimensional space following [3], combined with our proposed training-free method. The entire process requires only 10 seconds to one minute per scene (depending on scene scale), thanks to our effective cross-view feature aggregation and streaming updates at constant memory. For all experiments, we used an NVIDIA RTX 4090 GPU.

5.2. Analysis on LeRF-OVS dataset

Table 1 compares ours with state-of-the-art works on LERF-OVS in 2D and 3D. In 2D, per-view segmentation quality projected from 3D is checked, while in 3D, we directly assess multi-view consistent semantic reconstruction.

Quantitatives In 2D. Our method achieves the highest scores on both mIoU and mAcc, slightly surpassing the mIoU of Occam's LGS [3] and outperforming LangSplatV2 [19]. This improvement is consistent across diverse scenes, particularly in Figurines and Ramen, suggesting that our visibility-aware attribution reduces per-ray semantic noise without sacrificing fine-grained per-view accuracy. While GAGS [25] and LangSplat [28] also deliver competitive 2D scores, their performance drops with complex occlusions (*e.g.*, Ramen for GAGS), indicating that their 2D-driven assignments do not fully mitigate cross-view inconsistencies.

Quantitatives In 3D. The advantage of our method becomes more pronounced in 3D, with ours exceeding all baselines by a notable margin. The second best, CAGS [34], is a substantial 7.2 absolute mIoU points behind. The scene-level analysis reveals that our approach leads in Ramen, Teatime, and Waldo_Kitchen, and ranks second in Figurines, behind VoteSplat [12] due to its specialized multi-view voting. The gains are especially significant in large, cluttered environments (Teatime, Waldo_Kitchen), where our contribution-aware aggregation better preserves semantics despite severe occlusions.

The strong 3D consistency of our method contrasts with approaches like LangSplat and LEGaussian [31], whose high 2D accuracy does not translate to 3D performance, likely due to their lack of explicit handling of per-ray contribution and occlusion. Similarly, the post-hoc clustering methods OpenGaussian [38] and SuperGSeg [20] exhibit moderate 3D improvements but remain sensitive to upstream semantic drift, thereby limiting their robustness. Our performance relative to Occam's LGS (baseline) is note-

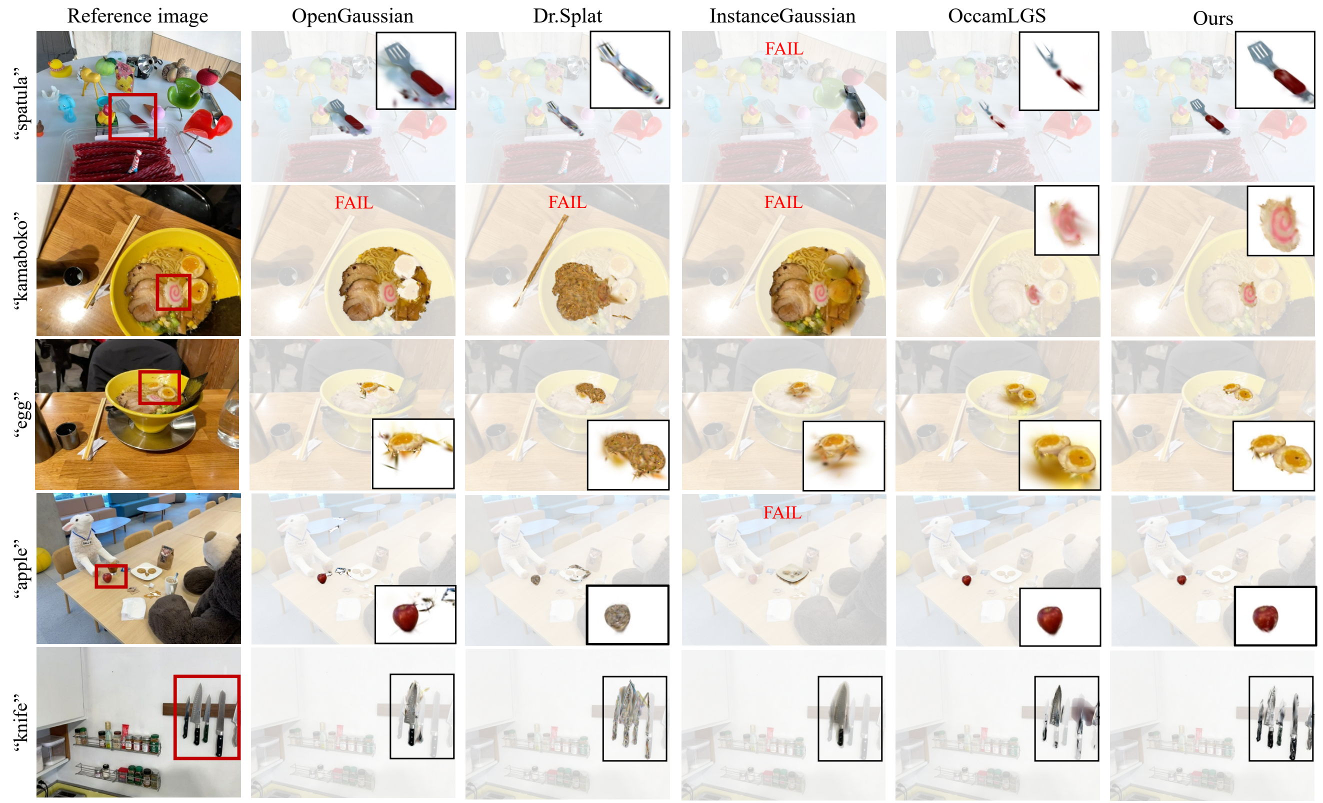


Figure 4. Qualitative 3D objects selections on LeRF-OVS [28]. We mark as failed those with low or zero IoU with the ground truth (red).

worthy: while both adopt streaming updates, our visibility-guided feature attribution yields much better performance in 3D, highlighting the effectiveness of improving the semantic assignment at the feature aggregation stage rather than solely relying on memory-efficient training.

Qualitatives in 3D. We show visual 3D results in Figure 4. Existing approaches, such as InstanceGaussian [18], frequently fail by retrieving incorrect objects across multiple scenes. This can be attributed to their reliance on appearance–semantic joint representations, which struggle to distinguish small objects with visually similar appearances. Clustering-based methods struggle with multiple instances that are closely related. For example, querying for *"knife"*, OpenGaussian [38] and InstanceGaussian [18] detect only one out of five knives, whereas Dr.Splat [13] and Occam's LGS [3] identify all knives but produce indistinct boundaries. In contrast, ours successfully localizes all knives with accurate and sharp delineations. Our approach also demonstrates robustness on challenging small-object queries, such as *"Kamaboko"* and *"egg"* in the *Ramen* scene. These targets lie within a heavily cluttered context (a bowl of ramen), making them particularly difficult to isolate. Competing methods [13, 18, 38] fail to recognize these objects, while Occam's LGS correctly retrieves them but with blurred contours. By comparison, ours produces precise boundaries and accurately captures fine object structures. Similar improvements are observed in the *"Spatula"* query, further illustrating that our visibility-aware gating not only mitigates occlusion effects but also enables the recovery of fine-grained details in complex scenes.

Method	19 classes		15 classes		10 classes	
	mIoU	mAcc	mIoU	mAcc	mIoU	mAcc
LangSplat [15]	2.45	8.59	3.45	13.21	6.48	21.89
OpenGaussian [38]	27.73	42.01	29.67	46.15	39.93	57.34
Dr. Splat [13]	29.31	47.68	33.25	54.33	44.19	65.19
Occam's LGS [3]	31.93	48.93	34.25	53.71	45.16	64.39
VALA [ours]	32.11	50.05	35.10	54.77	46.21	65.61

Table 2. Open-vocabulary 3D semantic segmentation task on the ScanNet-v2 dataset [5] across different amounts of classes.

5.3. 3D Semantic Segmentation on ScanNet

Quantitatives. As reported in Table 2, our method achieves the best performance across all evaluation settings, including the most challenging 19-class scenario. Compared to Occam's LGS [3], our contribution-aware aggregation is advantageous, demonstrating its ability to handle fine-grained class distributions. While Dr.Splat [13] attains competitive accuracy in reduced-category settings, it lags notably

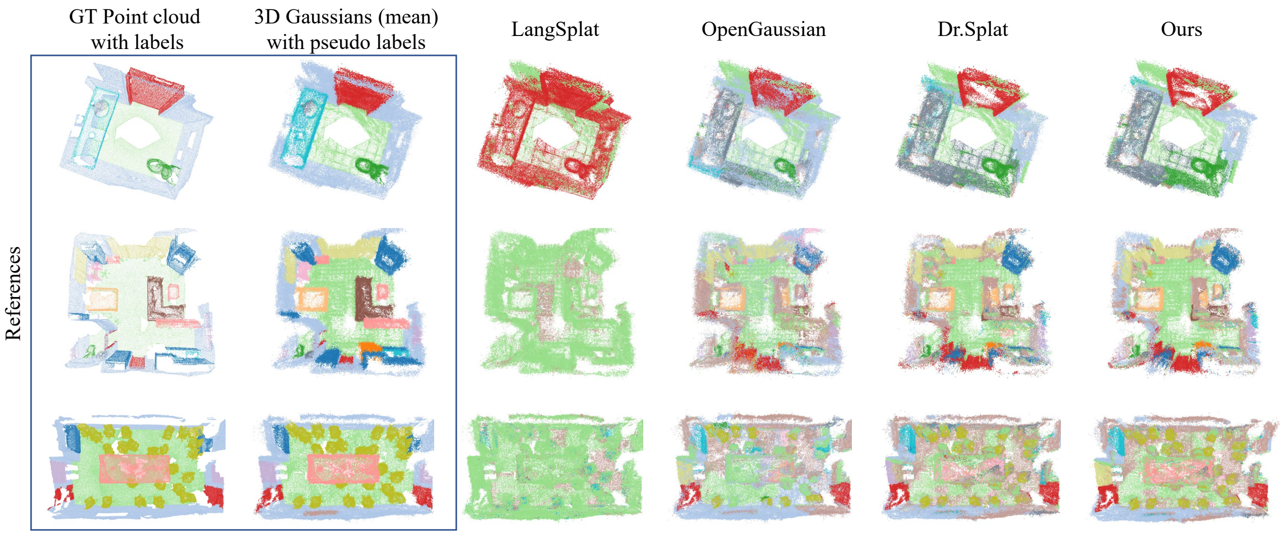

GT Point cloud with labels
3D Gaussians (mean) with pseudo labels
LangSplat
OpenGaussian
Dr.Splat
Ours
References

Figure 5. Qualitative results of 3D semantic segmentation with 19 classes on the ScanNet-v2 dataset [5].

in mIoU, indicating weaker spatial consistency. These results confirm that our method achieves robust and precise 3D segmentation across varying label granularities.

Qualitatives. Qualitative comparisons are presented in Figure 5. In the large and complex second room, our method accurately predicts the wall behind the bed (bed in orange), a structure often misclassified by others. In the smaller but more occluded third scene, our method also demonstrates superior 3D segmentation, capturing challenging objects such as the central table more effectively. This ability to recover occluded and fine-scale geometry is particularly beneficial for downstream applications such as 3D object localization. Overall, the qualitative results support the quantitative improvements, highlighting both the robustness and effectiveness of our proposed framework.

5.4. Ablation Study

We conduct an ablation study on LeRF-OVS [28], averaging the metrics over all scenes. Table 3 disentangles the contributions of our main components, namely visibility-aware gating and cosine-based geometric median. Starting from the baseline Occam's LGS [3], replacing the naive weighted mean with our cosine median (b) already improves performance, highlighting the advantage of robust aggregation in the embedding space. Incorporating visibility-aware gating further boosts results (c-d), where mass-coverage plus threshold gating (c) yields the strongest individual gain, while quantile pruning (d) provides complementary benefits. We also observe that our gating alone (f) is less effective compared to gating along with our robust median (VALA), showing that the precise aggregation is critical to fully exploit visibility cues. Lastly, we compare cosine and L1 (g) as median, with the former delivering superior results. Our full model (VALA) achieves the best overall performance, validating that both visibility-aware gating and cosine-based median aggregation are important for an accurate and view-consistent 2D-3D language lifting.

Ref.	Stage A	Stage B	Median	mIoU	mAcc
O.LGS [3]				47.22	74.86
(b)			cosine	49.03	80.08
(c)	✓		cosine	57.24	81.25
(d)		✓	cosine	55.21	80.37
VALA	✓	✓	cosine	**58.02**	**82.85**
(f)	✓	✓		52.29	76.17
(g)	✓	✓	L1	56.03	82.42

Table 3. Ablation on LeRF-OVS. First row is Occam's LGS [3], *i.e.*, our baseline. Stages from Section 4.1, Median from 4.2. All rows share the same data, rasterizer, and hyperparameters.

We refer to the **Supplementary Material** for additional details and results.

6. Conclusion

We introduced VALA, an efficient and effective method to address two fundamental problems in the feature aggregation of open-vocabulary recognition in 3DGS, namely (i) the propagation of 2D features to all Gaussians along a camera ray, and (ii) the multi-view inconsistency of semantic features. VALA tackles (i) with a visibility-aware distillation of language features based on a two-stage gating mechanism, and (ii) with a cosine variant of the geometric median, updating the features via streaming to keep the memory footprint low. These innovations ensure more appropriate features are assigned to the 3D Gaussians, ultimately leading to superior performance in open-vocabulary segmentation. Remarkably, the proposed VALA achieves state-of-the-art performance on 2D *and* 3D tasks on the reference datasets LeRF-OVS and ScanNet-v2.

References

[1] Amir Beck and Shoham Sabach. Weiszfeld's method: Old and new results. *Optimization Letters*, 9(1):1–18, 2015. See also preprint/PDF for historical notes. 5

[2] Cesar Cadena, Luca Carlone, Henry Carrillo, Yasir Latif, Davide Scaramuzza, José Neira, Ian Reid, and John J. Leonard. Past, present, and future of simultaneous localization and mapping: Toward the robust-perception age. *IEEE Transactions on Robotics*, 32(6):1309–1332, 2016. 1

[3] Jiahuan Cheng, Jan-Nico Zaech, Luc Van Gool, and Danda Pani Paudel. Occam's LGS: A simple approach for language Gaussian splatting. *arXiv preprint arXiv:2412.01807*, 2024. 1, 2, 3, 4, 6, 7, 8

[4] Christopher Choy, JunYoung Gwak, and Silvio Savarese. 4d spatio-temporal convnets: Minkowski convolutional neural networks. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 3075–3084, 2019. 1

[5] 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/CVF Conference on Computer Vision and Pattern Recognition*, 2017. 2, 5, 7, 8, 1, 3

[6] Ulrich Eckhardt. Weber's problem and weiszfeld's algorithm in general spaces. *Mathematical Programming*, 18(1):186–196, 1980. 5

[7] Francis Engelmann, Fabian Manhardt, Michael Niemeyer, Keisuke Tateno, and Federico Tombari. OpenNeRF: Open set 3D neural scene segmentation with pixel-wise features and rendered novel views. In *The Twelfth International Conference on Learning Representations*, 2024. 1, 2

[8] Andreas Geiger, Philip Lenz, and Raquel Urtasun. Are we ready for autonomous driving? the kitti vision benchmark suite. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 2443–2451, 2012. 1

[9] Xiuye Gu, Yen-Chun Kuo, Yin Cui, Zecheng Sun, David Zhang, and Steven C. H. Hoi. Open-vocabulary object detection via vision and language knowledge distillation. *arXiv preprint arXiv:2104.13921*, 2021. 1

[10] Shahram Izadi, David Kim, Otmar Hilliges, David Molyneaux, Richard Newcombe, Pushmeet Kohli, Jamie Shotton, Steve Hodges, Daniel Freeman, Andrew Davison, and Andrew Fitzgibbon. Kinectfusion: Real-time 3d reconstruction and interaction using a moving depth camera. In *ACM Symposium on User Interface Software and Technology (UIST)*, pages 559–568, 2011. 1

[11] Chao Jia, Yinfei Yang, Ye Xia, Yi-Ting Chen, Zarana Parekh, Hieu Pham, Quoc V. Le, Yunhsuan Sung, Zhen Li, and Thomas Duerig. Scaling up visual and vision-language representation learning with noisy text supervision. In *International Conference on Machine Learning (ICML)*, pages 4904–4916, 2021. 1

[12] Minchao Jiang, Shunyu Jia, Jiaming Gu, Xiaoyuan Lu, Guangming Zhu, Anqi Dong, and Liang Zhang. Votesplat: Hough voting gaussian splatting for 3d scene understanding. *arXiv preprint arXiv:2506.22799*, 2025. 1, 2, 4, 6

[13] 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 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2025. 1, 2, 3, 4, 6, 7

[14] Bernhard Kerbl, Georgios Kopanas, Thomas Leimkühler, and George Drettakis. 3D Gaussian Splatting for real-time radiance field rendering. *ACM Transactions on Graphics*, 42 (4):139–1, 2023. 1, 3

[15] 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

[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. 1, 2, 5, 6

[17] Georg Klein and David Murray. Parallel tracking and mapping for small ar workspaces. In *IEEE and ACM International Symposium on Mixed and Augmented Reality*, pages 225–234, 2007. 1

[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 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 14078–14088, 2025. 1, 2, 5, 6, 7

[19] Wanhua Li, Yujie Zhao, Minghan Qin, Yang Liu, Yuanhao Cai, Chuang Gan, and Hanspeter Pfister. Langsplatv2: High-dimensional 3d language gaussian splatting with 450+ fps. *arXiv preprint arXiv:2507.07136*, 2025. 6

[20] 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. 2, 4, 5, 6

[21] Horst Martini, Konrad J. Swanepoel, and Günter Weiss. On torricelli's geometrical solution to a problem of fermat. *Elemente der Mathematik*, 50(2):93–96, 1995. 5

[22] Ben Mildenhall, Pratul P Srinivasan, Matthew Tancik, Jonathan T Barron, Ravi Ramamoorthi, and Ren Ng. NeRF: Representing scenes as neural radiance fields for view synthesis. In *Proceedings of the European Conference on Computer Vision*, pages 405–421, 2020. 1, 2

[23] Raul Mur-Artal and Juan D. Tardós. Orb-slam2: An open-source slam system for monocular, stereo, and rgb-d cameras. *IEEE Transactions on Robotics*, 33(5):1255–1262, 2017. 1

[24] Songyou Peng, Kyle Genova, Chiyu Max Jiang, Andrea Tagliasacchi, Marc Pollefeys, Thomas Funkhouser, Matthias Nießner, and Sida Peng Liu. Openscene: 3d scene understanding with open vocabularies. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 1786–1796, 2023. 1

[25] 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

[26] Jens Piekenbrinck, Christian Schmidt, Alexander Hermans, Narunas Vaskevicius, Timm Linder, and Bastian Leibe. Opensplat3d: Open-vocabulary 3d instance segmentation using gaussian splatting. In *Proceedings of the Computer Vision and Pattern Recognition Conference*, pages 5246–5255, 2025. 2, 5

[27] Charles R Qi, Hao Su, Kaichun Mo, and Leonidas J Guibas. PointNet: Deep learning on point sets for 3D classification and segmentation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 652–660, 2017. 1

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

[29] 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 ACM International Conference on Multimedia*, pages 5328–5337, 2024. 2, 6

[30] 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 *Proceedings of the International Conference on Machine Learning*, pages 8748–8763. PMLR, 2021. 1, 2, 6

[31] 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. 2, 6

[32] Pei Sun, Henrik Kretzschmar, Xerxes Dotiwalla, Aurelien Chouard, Sergio Casas, Wenjie Lin, Abbas Sadat, Balakrishnan Varadarajan, Jonathon Shlens, Zhifeng Chen, Alan Yuille, and Dragomir Anguelov. Scalability in perception for autonomous driving: Waymo open dataset. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 2443–2451, 2020. 1

[33] Pei Sun, Henrik Kretzschmar, Xerxes Dotiwalla, Aurelien Chouard, Vijaysai Patnaik, Paul Tsui, James Guo, Yin Zhou, Yuning Chai, Benjamin Caine, Vijay Vasudevan, Wei Han, Jiquan Ngiam, Hang Zhao, Aleksei Timofeev, Scott Ettinger, Maxim Krivokon, Amy Gao, Aditya Joshi, Yu Zhang, Jonathon Shlens, Zhifeng Chen, and Dragomir Anguelov. Scalability in perception for autonomous driving: Waymo open dataset. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2020. 4

[34] Wei Sun, Yanzhao Zhou, Jianbin Jiao, and Yuan Li. Cags: Open-vocabulary 3d scene understanding with context-aware gaussian splatting. *arXiv preprint arXiv:2504.11893*, 2025. 1, 6

[35] Hugues Thomas, Charles R. Qi, Jean-Emmanuel Deschaud, Beatriz Marcotegui, François Goulette, and Leonidas J. Guibas. Kpconv: Flexible and deformable convolution for point clouds. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 6410–6419, 2019. 1

[36] 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. 2

[37] Endre Weiszfeld and Frank Plastria. On the point for which the sum of the distances to n given points is minimum. *Annals of Operations Research*, 167(1):7–41, 2008. 5

[38] 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. 1, 2, 4, 5, 6, 7

[39] Vickie Ye, Ruilong Li, Justin Kerr, Matias Turkulainen, Brent Yi, Zhuoyang Pan, Otto Seiskari, Jianbo Ye, Jeffrey Hu, Matthew Tancik, and Angjoo Kanazawa. gsplat: An open-source library for gaussian splatting. *Journal of Machine Learning Research*, 26(34):1–17, 2025. 6, 1

[40] Chenlu Zhan, Yufei Zhang, Gaoang Wang, and Hongwei Wang. Hi-lsplat: Hierarchical 3d language gaussian splatting. *arXiv preprint arXiv:2506.06822*, 2025. 2

[41] 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. 2

[42] Xingyi Zhou, Rohit Girdhar, Armand Joulin, Philipp Krähenbühl, and Ishan Misra. Detecting twenty-thousand classes using image-level supervision. In *European Conference on Computer Vision*, pages 350–368, 2022. 1

[43] 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. 2