SuperGSeg: Open-Vocabulary 3D Segmentation with Structured Super-Gaussians

Siyun Liang[1,2*] Sen Wang[1,4*] Kunyi Li[1,4] Michael Niemeyer[3] Stefano Gasperini[1,4,5]
Hendrik P.A. Lensch[2] Nassir Navab[1] Federico Tombari[1,3]
[1]Technical University of Munich [2]University of Tübingen
[3]Google [4]Munich Center for Machine Learning [5]VisualAIs

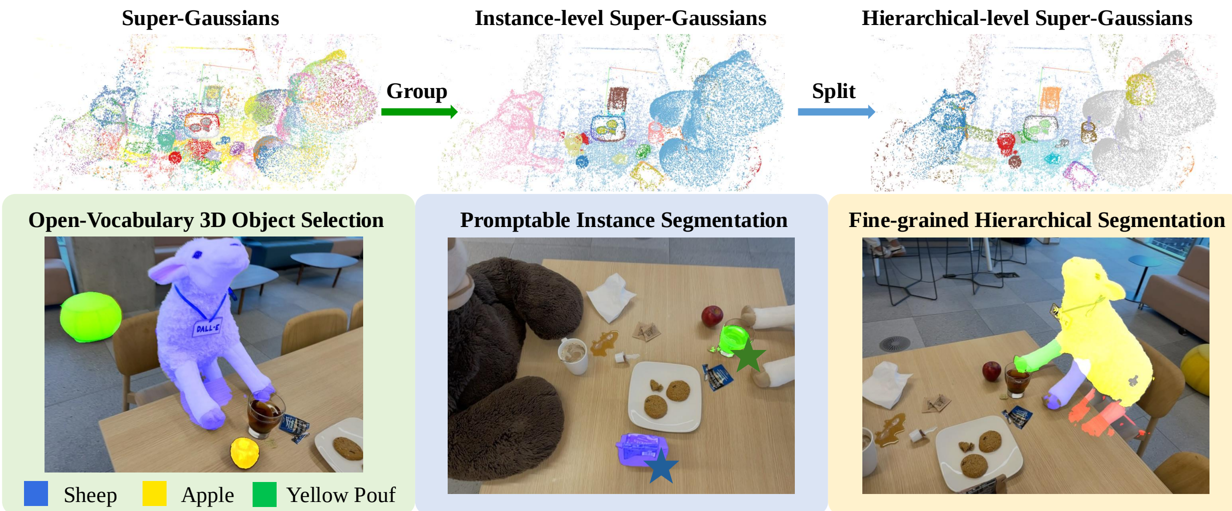

Figure 1. We present **SuperGSeg**, a novel method that clusters similar Gaussians into superpoint-like representations, termed Super-Gaussians (SuperGs). SuperGSeg enables efficient integration of diverse feature fields for comprehensive 3D scene understanding. Left: Querying SuperGs' language features enables open-vocabulary 3D object selection, producing consistent 3D masks that extend beyond 2D visible surfaces, e.g., the leg of the sheep under the table. Middle: Grouping SuperGs by instance features enables promptable instance segmentation. Right: Further splitting instances via hierarchical features enables fine-grained hierarchical segmentation.

Abstract

3D Gaussian Splatting has recently gained traction for its efficient training and real-time rendering. While its vanilla representation is mainly designed for view synthesis, recent works extended it to scene understanding with language features. However, storing additional high-dimensional features per Gaussian for semantic information is memory-intensive, which limits their ability to segment and interpret challenging scenes. To this end, we introduce SuperGSeg, a novel approach that fosters cohesive, context-aware hierarchical scene representation by disentangling segmentation and language field distillation. SuperGSeg first employs neural 3D Gaussians to learn geometry, instance and hierarchical segmentation features from multi-view images with the aid of off-the-shelf 2D masks. These features are then leveraged to create a sparse set of Super-Gaussians. Super-Gaussians facilitate the lifting and distillation of 2D language features into 3D space. They enable hierarchical scene understanding with high-dimensional language feature rendering at moderate GPU memory costs. Extensive experiments demonstrate that SuperGSeg achieves remarkable performance on both open-vocabulary object selection and semantic segmentation tasks. More results at `supergseg.github.io`.

1. Introduction

3D Gaussian Splatting (3DGS) [1] has rapidly emerged as a compelling alternative to NeRF [2] for its efficient training, real-time rendering, and explicit 3D representation. These advantages make 3DGS well-suited for a broad range of ap-

* Equal contribution.

plications, including 3D reconstruction [3–5], content generation [6], and scene understanding [7–12]. A particularly promising direction involves extending 3DGS frameworks to open-vocabulary understanding, enabling flexible, language-driven interaction with 3D scenes [13, 14].

Several recent methods aim to enable such open-vocabulary capabilities in 3DGS by distilling language features from both 2D [7, 9, 15, 16] and 3D [11, 12] perspectives. In 2D-based methods, language features extracted from images are lifted into 3D by exploiting the multi-view consistency inherent in 3DGS rendering. To reduce the substantial memory and computation overhead of storing and processing high-dimensional language features for each Gaussian, these methods employ dimensionality reduction techniques [7, 9]. However, this compression inevitably discards fine-grained semantic information. Another limitation is their inability to recognize partially occluded objects, which is often necessary in 3D understanding tasks. Text queries are performed on rendered pixels, which only capture the visible surface along each viewing ray. Consequently, objects that are partially or fully hidden cannot be retrieved. In contrast, 3D methods [11, 12] perform text queries directly in 3D space at the point level, which enables the retrieval of occluded objects by rendering the queried Gaussians into masks (see Figure 5), but also introduces new limitations. By directly associating language features with individual Gaussians and decoupling alpha blending, they cannot render consistent language feature maps in pixel space, which in turn makes them unsuitable for tasks such as pixel-wise dense semantic segmentation in 2D.

To address the aforementioned issues, we introduce a novel approach that: (1) preserves high-dimensional language feature embeddings without information loss, (2) handles occlusions by operating directly in 3D space, and (3) supports multi-granular segmentation, ultimately enabling open-vocabulary queries in both 2D and 3D, as shown in Figure 1. Inspired by superpoints [17] in point cloud analysis, our method clusters millions of Gaussians into a compact set of Super-Gaussians (SuperGs). However, due to the inherent noise in Gaussian point clouds, clustering solely based on Gaussian positions often produces suboptimal groupings. Instead, we leverage instance and hierarchical features extracted from grouped SAM masks [18] to guide clustering via an adaptive online clustering network [19]. For open-vocabulary scene understanding, we further distill 2D CLIP features [13] onto SuperGs that integrate both spatial and semantic information. This compact representation allows language features to be assigned at the SuperG level rather than to each individual Gaussian [7–9], thereby reducing the number of learnable language features from millions to only thousands, significantly lowering memory usage while retaining the full descriptive power of the original high-dimensional features.

Extensive experiments on the LERF-OVS [7] and ScanNet [20] datasets show that our method achieves remarkable performance in open-vocabulary 3D object retrieval and scene-level semantic segmentation, demonstrating superior capability in producing complete and consistent masks for 3D object retrieval and capturing fine-grained scene details for 2D dense pixel-wise segmentation. We summarize the main contributions as follows:

- We introduce SuperGSeg, a novel 3D scene understanding framework built on Super-Gaussian representations, enabling effective high-dimensional language feature distillation without information loss.
- We propose a novel neural Gaussian rasterization pipeline that distills instance and hierarchical feature fields, facilitating Super-Gaussian clustering and supporting multi-granular scene understanding.
- We design an online clustering network that adaptively fuses geometric, semantic, and appearance cues to generate Super-Gaussians, thus improving clustering quality.

2. Related Work

3D Open-Vocabulary Understanding. Advancements in universal 2D scene understanding, driven by foundation models such as CLIP [13] and SAM [21], have motivated the integration of language-aligned features into 3D scene representations. Early efforts incorporated these 2D features [13, 22] into NeRF-based representations [23, 24], enabling open-vocabulary queries in 3D scenes but at the cost of slow rendering and high memory usage. More recently, the emergence of 3DGS as a high-quality, real-time alternative for novel view synthesis has inspired extensions toward 3D scene understanding. For example, LangSplat [7] employs a scene-specific language autoencoder to compress high-dimensional CLIP features, providing clear object boundaries in rendered feature images while reducing memory usage. Feature3DGS [9] introduces a parallel Gaussian rasterizer with a lightweight convolutional decoder to distill high-dimensional features for tasks like scene editing and segmentation. However, these dimensionality reduction techniques inevitably discard fine-grained semantic information. OpenGaussian [11] instead directly associates uncompressed, lossless CLIP features with 3D Gaussians, preserving complete semantics and enabling the retrieval of visually occluded objects by performing queries directly in 3D space. Nevertheless, its decoupled language codebook design makes per-pixel 2D language feature rendering infeasible, thereby limiting performance on dense, pixel-wise semantic prediction tasks.

Despite notable progress, most existing methods focus primarily on instance-level knowledge while neglecting fine-grained part-level semantics [10, 11], or require separate models for different semantic granularities [7]. While recent methods [18, 25] explore hierarchical 3D un-

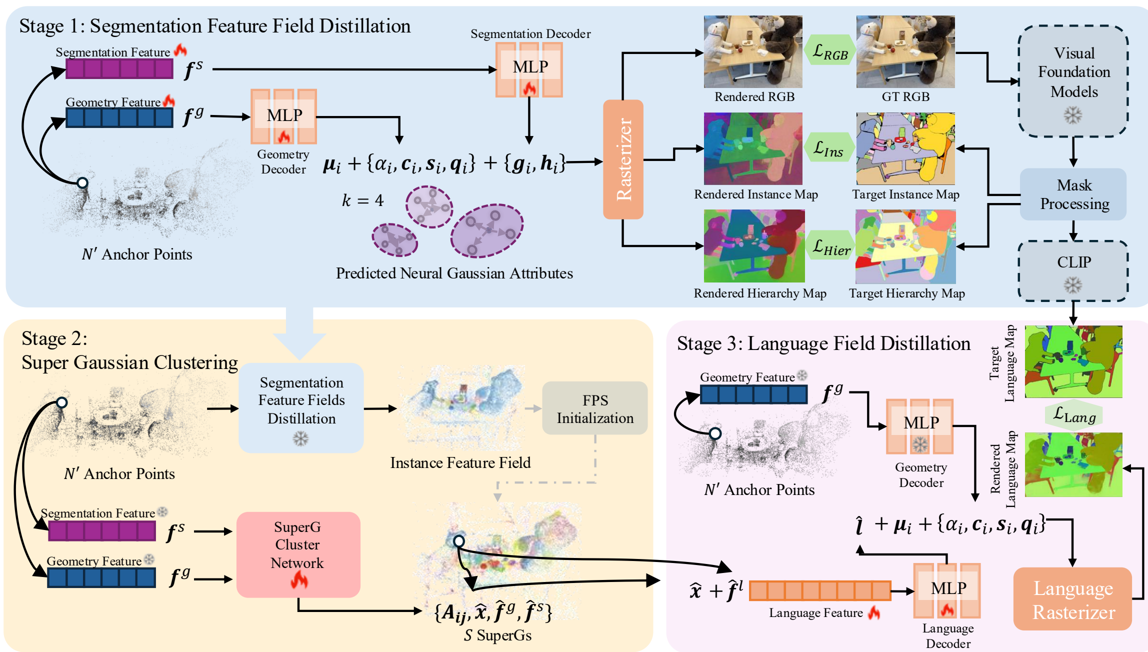

Figure 2. **SuperGSeg Overview.** We initialize the 3D Gaussians from a sparse set of anchor points, each generating k Gaussians with corresponding attributes. First, we train the appearance and segmentation features using RGB images and segmentation masks generated by SAM [21]. Next, we use the segmentation features and their spatial positions to produce a sparse set of Super-Gaussians, each carrying a 512-dimensional language feature. Finally, we train this high-dimensional language feature using a 2D feature map from CLIP [13].

derstanding at the part-level, they lack support for open-vocabulary language queries, leaving the joint modeling of multi-granularity 3D representation with language feature largely unexplored. In contrast, our method integrates both instance and hierarchical features from 2D inputs, and introduces a Super-Gaussian based language field that fuses segmentation information with the spatial distribution of 3D Gaussians, thereby enabling open-vocabulary, multi-granularity, and occlusion-robust 3D segmentation.

Superpoints. Superpoints have long served as fundamental primitives for various point cloud understanding tasks [19, 26–31]. Early approaches, such as Voxel Cloud Connectivity Segmentation (VCCS) [32], segment a voxelized 3D grid into spatially coherent regions using region-growing variants of K-means clustering. More recent works leverage learned point cloud representations [33, 34] to infer superpoints directly from 3D scans [19, 27, 35]. Superpoints have also been adopted for open-vocabulary 3D segmentation [17], demonstrating robustness in complex scenes. However, directly applying superpoint methods to 3DGS is challenging due to noisy Gaussian geometry. To address this, we leverage instance- and part-level cues from 2D foundation models to guide superpoint formation, effectively bridging high-quality 2D features with noisy 3D Gaussian representations.

3. Method

Given a set of posed RGB images, our goal is to reconstruct a 3D scene with a compact language feature field that supports open-vocabulary querying of arbitrary concepts. To achieve this, we propose a three-stage training paradigm, as shown in Figure 2. In the first stage, we train a neural variant of 3DGS [36] to reconstruct scene geometry using N' anchor points, each having a geometry feature $\boldsymbol{f}^g$ and a segmentation feature $\boldsymbol{f}^s$. Anchor points are then spawned into a set of neural Gaussians and optimized. In the second stage, a learnable cluster network groups the anchors into S SuperGs using $\boldsymbol{f}^g$, $\boldsymbol{f}^s$, and anchor position $\boldsymbol{x}$, ensuring geometric and semantic consistency. Since $S \ll N'$, this yields a far more compact representation. In the third stage, we learn a language feature $\widehat{\boldsymbol{f}}^l$ for each SuperG, enabling open-vocabulary queries on just S SuperGs rather than millions of individual Gaussians.

3.1. Preliminaries: Neural Gaussian Splatting

We begin with Stage 1 of our pipeline: modeling the scene geometry with Scaffold-GS [36] structure. Vanilla 3DGS represents a scene with N Gaussians, each parameterized by a center $\boldsymbol{\mu}$, opacity α, color $\boldsymbol{c}$, scale $\boldsymbol{s}$ and quaternion $\boldsymbol{q}$. These Gaussians are projected onto the image plane [37] and rendered into RGB images via α-blending. While achieving leading rendering quality and speed, optimizing each Gaussian independently often leads to overfitting, redundancy, and degraded robustness in challenging regions such as texture-less surfaces. Scaffold-GS addresses these issues by voxelizing the scene into N' anchor points, each at position $\boldsymbol{x}$. From each anchor, k neural Gaussians are derived, where centers are computed as $\boldsymbol{x}$ plus learnable offsets, and the remaining attributes $(\alpha, \boldsymbol{c}, \boldsymbol{s}, \boldsymbol{q})$ are produced on the fly from the anchor's geometry feature $\boldsymbol{f}^g$ via dedicated MLPs. By tying Gaussians to anchors, Scaffold-GS constrains their spatial distribution to the scene structure, preventing uncontrolled growth and improving robustness.

Training in 3DGS typically relies on a photometric loss $\mathcal{L}_{RGB}$, where rendered RGB images are supervised against ground-truth views. Unlike vanilla 3DGS that optimizes $(\boldsymbol{\mu}, \alpha, \boldsymbol{c}, \boldsymbol{s}, \boldsymbol{q})_N$, with N often reaching millions for complex scenes, Scaffold-GS optimizes only $(\boldsymbol{f}^g)_{N'}$, the Gaussian offsets, and MLP weights, which significantly reduces parameters. This anchor-based formulation naturally yields a coarse partition of the Gaussian space, providing a strong basis for our subsequent clustering into SuperGs.

3.2. Segmentation Feature Field Distillation

Given N' anchor points representing the scene geometry, the next step is to group them into S superpoints, each forming a SuperG through its derived neural Gaussians. Ideally, each SuperG should align with a single semantic entity in the scene. However, clustering anchors solely by their geometry features $\boldsymbol{f}^g$ or positions $\boldsymbol{x}$ is suboptimal, since anchors from distinct objects can be spatially adjacent or geometrically similar. To overcome this limitation, we introduce an additional segmentation feature $\boldsymbol{f}^s$, distilled from 2D SAM masks, which encodes both instance- and part-level semantic cues to guide the SuperG clustering.

Hierarchical Partitioning of SAM Masks. Given an input RGB image, SAM [21] generates a set of 2D segmentation masks. These masks can, however, overlap with each other, leading to pixels belonging to multiple masks and thus obscuring the inherent part-instance hierarchy. Prior works either train separate models for each mask level [7, 38, 39], which is less efficient, or rely only on coarse instance-level masks [11, 40], discarding the finer part-instance relations. To overcome this, we adopt a hierarchical representation [18] that restructures the masks into non-overlapping instance-level masks $\mathcal{M}$ for whole objects and part-level patches $\mathcal{P}$ for finer components, which together provide supervision for learning both object-level semantics and intra-object details in the segmentation feature field. Implementation details and example mask visualizations are provided in Appendix B.

Instance and Hierarchical Feature Field. As shown in Figure 2, we assign each anchor point a segmentation feature $\boldsymbol{f}^s$. We pass $\boldsymbol{f}^s$ together with the anchor position $\boldsymbol{x}$

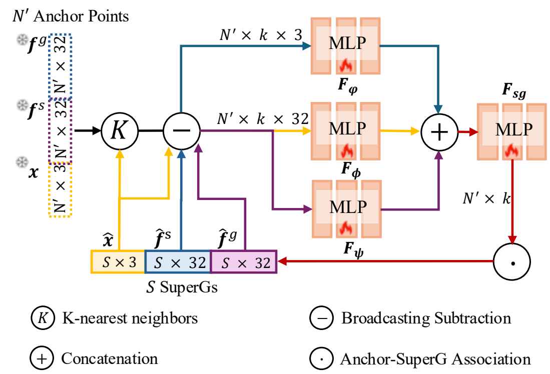

Figure 3. The architecture of the SuperG Cluster Network.

to a segmentation decoder to get the instance feature $\boldsymbol{g}$ and hierarchical feature $\boldsymbol{h}$ for each neural Gaussian. Through the vanilla Gaussian Splatting pipeline, we rasterize $\boldsymbol{g}$ and $\boldsymbol{h}$ to generate the 2D instance feature map $\hat{\boldsymbol{G}} \in \mathbb{R}^{D_g \times H \times W}$ and the 2D hierarchical feature map $\hat{\boldsymbol{H}} \in \mathbb{R}^{D_h \times H \times W}$.

To train the segmentation features, we leverage a contrastive learning objective [18, 41] to enforce cross-view consistency, encouraging features from the same mask to be similar while pushing apart those from different masks. Specifically, we represent the set of SAM-generated instance-level masks as $\mathcal{M} = \{\boldsymbol{m}^p \in \mathbb{R}^{H \times W} \mid p = 1, \ldots, |\mathcal{M}|\}$. Given an instance mask $\boldsymbol{m}^p$, we collect all rendered instance features whose pixels fall inside the mask, and denote this set as $\hat{\boldsymbol{g}}^p = \{\hat{\boldsymbol{g}}^p_t \in \hat{\boldsymbol{G}} \mid t = 1, \cdots, |\hat{\boldsymbol{g}}^p|\}$. We compute the mean instance feature value within $\boldsymbol{m}^p$ as $\bar{\boldsymbol{g}}^p$ and the contrastive instance feature loss $\mathcal{L}_{Ins}$ is:

$$\mathcal{L}_{Ins} = -\frac{1}{|\mathcal{M}|} \sum_{p=1}^{|\mathcal{M}|} \sum_{t=1}^{|\hat{\boldsymbol{g}}^p|} \log \frac{\exp(\hat{\boldsymbol{g}}^p_t \cdot \bar{\boldsymbol{g}}^p / \tau_p)}{\sum_{q=1}^{|\mathcal{M}|} \exp(\hat{\boldsymbol{g}}^p_t \cdot \bar{\boldsymbol{g}}^q / \tau_q)}, \quad (1)$$

where τ is the cluster temperature. We adopt a similar hierarchical feature loss $\mathcal{L}_{Hier}$ from Omniseg3D [18], but applied to part-level patches $\mathcal{P}$ to supervise our hierarchical feature $\boldsymbol{h}$. We refer to Appendix B for more details. Combined with the reconstruction loss introduced in Section 3.1, these objectives define the overall training loss for Stage 1:

$$\mathcal{L}_{stage1} = \mathcal{L}_{RGB} + \lambda_{Ins}\mathcal{L}_{Ins} + \lambda_{Hier}\mathcal{L}_{Hier}. \quad (2)$$

3.3. Super-Gaussian Clustering

After learning anchor-level geometry and segmentation features, we proceed to Stage 2, where anchors are grouped into semantically meaningful SuperGs to form a compact representation. However, contrastive learning struggles to separate objects that never co-occur in training [25], potentially grouping too distant Gaussians. To ensure spatial compactness and semantic consistency, we incorporate the anchor positions $\boldsymbol{x}$ alongside segmentation features $\boldsymbol{f}^s$, while geometric features $\boldsymbol{f}^g$ provide appearance cues for refinement. A straightforward baseline is to apply K-means clustering [12] to the concatenated feature space of $\{\boldsymbol{x}, \boldsymbol{f}^g, \boldsymbol{f}^s\}$. Yet, this approach fails when appearance cues misalign with semantics (e.g., diverse textures within an object). Moreover, K-means assumes equal importance across concatenated features, without the flexibility to adapt their relative relevance during clustering.To improve the clustering quality, we instead propose a learnable SuperG clustering network (see Figure 3), inspired by [19]. It follows two steps: initialization and iterative refinement.

Super-Gaussian Initialization. We apply the Farthest Point Sampling algorithm [42] on anchor points to initialize SuperGs, averaging each a position $\widehat{\boldsymbol{x}}$. Each SuperG has a geometry feature $\widehat{\boldsymbol{f}}^g$ and segmentation feature $\widehat{\boldsymbol{f}}^s$, which are initialized as the mean value of the corresponding anchors' features $\{\boldsymbol{f}^g, \boldsymbol{f}^s\}$.

Super-Gaussian Update. We denote the nearest k SuperGs to the i-th anchor as $\mathcal{N}_i$. The association probability matrix $\boldsymbol{A} \in \mathbb{R}^{N' \times k}$ [19, 43] is used to weight the contribution of each SuperG to its corresponding anchor, where N' is the number of anchors and k is the number of nearest SuperGs. Specifically, the association probability between the j-th SuperG ($j \in \mathcal{N}_i$) and the i-th anchor is:

$$\boldsymbol{A}_{ij} = F_{sg}\left(F_\phi(\boldsymbol{x}_i, \widehat{\boldsymbol{x}}_j), F_\varphi(\boldsymbol{f}_i^s, \widehat{\boldsymbol{f}}_j^s), F_\psi(\boldsymbol{f}_i^g, \widehat{\boldsymbol{f}}_j^g)\right), \quad (3)$$

where F_ϕ, F_φ, and F_ψ are lightweight MLP decoders that output relevance weights in terms of spatial, semantic, and geometric information, respectively. The concatenated weights are then passed to the prediction decoder F_{sg} for the normalized association probability matrix prediction. Unlike K-means, this design dynamically adjusts the contribution of each SuperG to its corresponding anchor.

We iteratively update SuperGs through the association matrix $\boldsymbol{A}$. At iteration $t+1$, each SuperG's position and features are updated with its corresponding anchors:

$$\widehat{\mathbf{e}}_j^{t+1} = \frac{1}{\sum_{i=1}^{N'} \mathbb{I}(j \in \mathcal{N}_i)\boldsymbol{A}_{ij}^t} \sum_{i=1}^{N'} \mathbb{I}(j \in \mathcal{N}_i)\boldsymbol{A}_{ij}^t \mathbf{e}_i, \quad (4)$$

where $\mathbb{I}$ denotes the indicator function, $\mathbf{e} \in \{\boldsymbol{x}, \boldsymbol{f}^g, \boldsymbol{f}^s\}$ are the anchor's attributes and $\widehat{\mathbf{e}} \in \{\widehat{\boldsymbol{x}}, \widehat{\boldsymbol{f}}^g, \widehat{\boldsymbol{f}}^s\}$ are SuperG's.

We optimize the SuperG clustering network to learn the association matrix $\boldsymbol{A}$, ensuring that the derived SuperG attributes $\widehat{\mathbf{e}}$ accurately reconstruct the anchor attributes $\mathbf{e}$. Note that $\mathbf{e}$ from Stage 1 (Section 3.2) are now frozen:

$$\mathcal{L}_{recon,\mathbf{e}} = \frac{1}{N'} \sum_{i=1}^{N'} \|\mathbf{e}_i - \sum_{j \in \mathcal{N}_i} \boldsymbol{A}_{ij} \widehat{\mathbf{e}}_j\|. \quad (5)$$

However, anchors within the same SuperG may be semantically similar yet spatially distant, especially when contrastive learning fails to optimize instances that never co-occur in the same view. To enforce spatial coherence, we introduce a compactness objective:

$$\mathcal{L}_{compact,\mathcal{X}} = \frac{1}{S} \sum_{j=1}^{S} \sum_{\mathbf{x} \in \mathcal{X}_j} \|\mathbf{x} - \widehat{\boldsymbol{x}}_j\|, \quad (6)$$

where $\mathcal{X}_j$ is the set of anchors' position assigned to the j-th SuperG. This loss encourages assigned anchors to cluster around their SuperG center and avoid fragmentation.

3.4. Language Field Distillation

Building on the clustering from Stage 2 (Section 3.3), in Stage 3, we distill 2D CLIP features into our compact set of S SuperGs, rather than into millions of individual 3D Gaussians to enable open-vocabulary 3D scene understanding. This design ensures consistent, robust, and high-dimensional language representations, while avoiding the feature degradation typically caused by the lossy compression used in Gaussian-based distillation approaches.

Since all Gaussians within a SuperG are expected to share the same semantics, we assign each SuperG a learnable latent language feature $\widehat{\boldsymbol{f}}^l$. As shown in Figure 2, this latent feature, together with the SuperG position $\widehat{\boldsymbol{x}}$, is decoded by a language feature MLP F_L to produce a CLIP-aligned feature: $\widehat{\boldsymbol{l}} = F_L(\widehat{\boldsymbol{f}}^l, \widehat{\boldsymbol{x}})$. We then modify the rasterizer to render a language feature map $\hat{\boldsymbol{L}}$, using $\widehat{\boldsymbol{l}}$ and the anchor-SuperG association map $\boldsymbol{A}$. For supervision, instance masks obtained in Section 3.2 are encoded using the CLIP image encoder to produce target 2D CLIP features $\boldsymbol{L}$. The latent features $\widehat{\boldsymbol{f}}^l$ and the decoder F_L are jointly optimized using a cosine similarity loss:

$$\mathcal{L}_{Lang} = 1 - \cos(\hat{\boldsymbol{L}}, \boldsymbol{L}). \quad (7)$$

4. Experiments

4.1. Experimental Setup

Datasets. We evaluate our method on the **open-vocabulary novel view semantic segmentation and object selection tasks** using the ScanNet v2 [20] and LERF-OVS [7] datasets. ScanNet v2 [20] includes posed RGB images and 2D semantic labels of indoor scenes. We randomly select 8 scenes from the dataset. These include a variety of indoor environments, e.g., living rooms, bedrooms, kitchens, and offices. For each scene, we split the data into a training set (composed of every 20th image from the original sequence) and a test set (derived from the intermediate images between the training set samples). For semantic segmentation, we specifically use the 20 object categories. LERF-OVS [7] consists of complex in-the-wild scenes captured with consumer-level devices, annotated with ground truth masks of textual queries to enable evaluation for open-vocabulary object selection tasks.

Baselines and Metrics. We compare our method with representative NeRF-based and 3DGS-based baselines, including LERF [23], LangSplat [7], LEGaussian [8], and

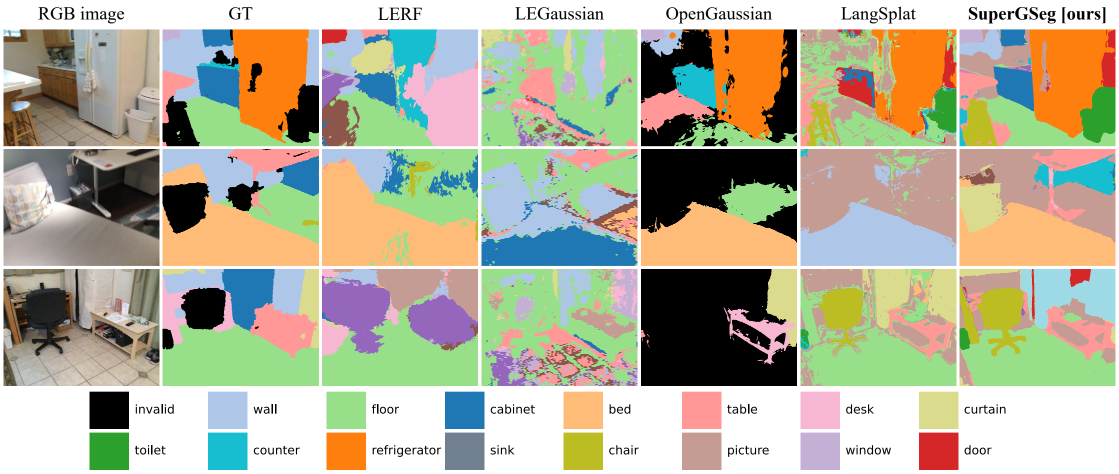

Figure 4. Qualitative comparison of semantic segmentation predictions on the ScanNet v2 dataset [20].

Method	**mean**		*wall*		*floor*		*cabinet*		*chair*		*refrigerator*		*curtain*	
	mIoU	mAcc	mIoU	mAcc	mIoU	mAcc	mIoU	mAcc	mIoU	mAcc	mIoU	mAcc	mIoU	mAcc
LERF [23]	38.5	60.4	35.2	82.8	60.1	68.8	52.0	82.7	10.9	10.9	69.9	90.2	70.2	77.8
LEGaussians [8]	8.7	33.2	17.9	53.1	14.6	20.6	2.7	18.6	0.4	28.7	9.0	74.3	1.9	10.4
OpenGaussian [11]	24.1	68.7	13.4	**96.6**	31.2	74.4	0.3	22.9	36.5	83.4	**88.0**	**98.3**	17.7	**79.2**
LangSplat [7]	27.6	48.3	45.3	72.6	43.3	45.6	24.8	56.7	18.0	48.5	0.7	33.3	46.8	66.5
SuperGSeg [ours]	**54.7**	**74.7**	**58.8**	92.9	**53.6**	**86.5**	**69.8**	**83.8**	**80.4**	**83.8**	79.4	80.2	**61.8**	64.5

Table 1. Comparison on the ScanNet v2 dataset [20]. We report the mean result and detailed scores for the most common object categories, following the evaluation protocol of [44]. Results for more categories are provided in the Appendix E.

OpenGaussian [11]. For the open-vocabulary semantic segmentation task, CLIP-encoded text features are compared with rendered 2D language feature maps via cosine similarity to produce per-pixel semantic predictions [9], evaluated with mean Intersection over Union (mIoU) and mean Accuracy (mAcc). For the open-vocabulary object selection task, we perform text queries directly in 3D space [11], retrieving the most relevant SuperGs and rendering them into 2D for evaluation with mIoU and mAcc. Since NeRF is an implicit representation without explicit 3D positions, LERF cannot be applied to this task. We also report inference-time efficiency, measuring both runtime and memory consumption for text queries on trained 3D scenes. Specifically, we perform multiple queries from different viewpoints and report the average query time. We consider this metric particularly important for assessing the feasibility of deploying models on resource-constrained devices and enabling real-time querying in practical scenarios.

Implementation Details. The training process is divided into 3 stages. In the first stage, we train the Scaffold-GS [36] with instance and hierarchical features for 30k iterations. In the second stage, we freeze the geometry and segmentation features from stage one and train only the SuperG clustering network for another 30k iterations. In the last stage, we freeze all other parameters and optimize the language features for each SuperG for 10k iterations. For more implementation details, we refer to Appendix B.

4.2. Open-Vocabulary Semantic Segmentation

Quantitative Results. As shown in Table 1, SuperGSeg achieves the best overall scores in both mIoU and mAcc among the compared methods, demonstrating its effectiveness in capturing the open-set information of the scene, yielding remarkable performance in a variety of object categories. In comparison, LEGaussian [16] shows lower performance on both metrics, suggesting limited generalization across multiple object categories. LangSplat [7] performs better than LEGaussian but still shows reduced accuracy in more diverse categories. OpenGaussian [11] obtains competitive results on certain large structures such as wall and floor, but its overall scene-level performance remains below ours. LERF [23] achieves the second-highest mIoU, though its relatively low mAcc suggests difficulties in producing clear segmentation boundaries.

Qualitative Results. As shown in Figure 4, our method produces sharper and more semantically consistent masks than the compared methods. While OpenGaussian [11] demonstrates competitive performance in 3D object-level

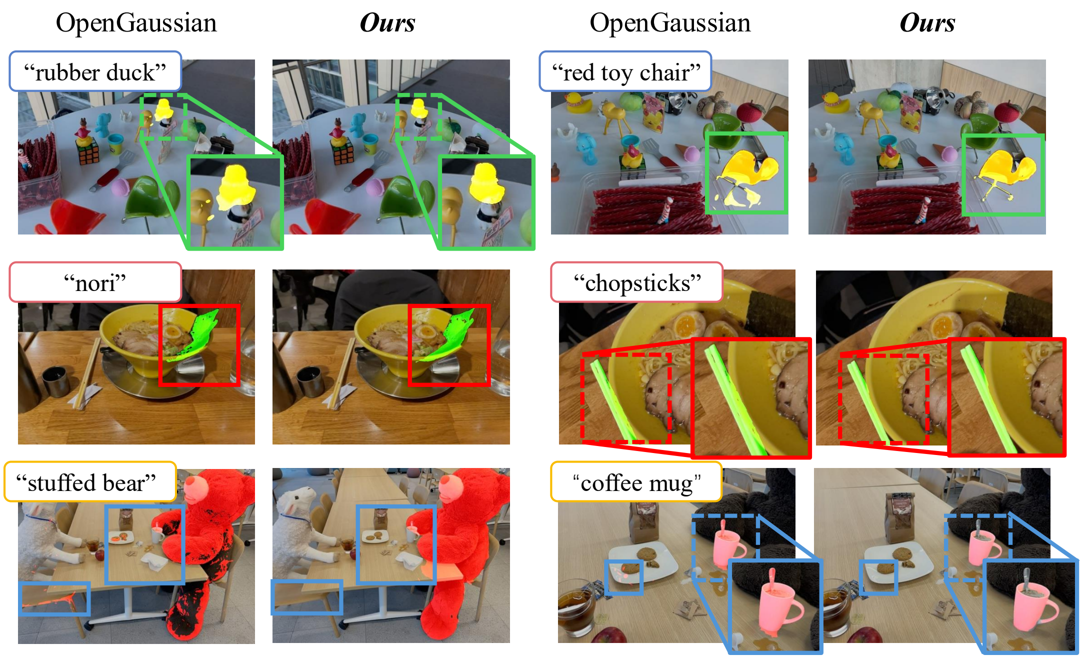

Figure 5. Qualitative comparison on the LERF-OVS dataset [23] for the open-vocabulary 3D object selection task. Text queries for each scene are displayed in quotation marks. SuperGSeg delivers more precise and less noisy segmentation masks.

Method	Inference Time	Inference Mem.	**mean** mIoU	**mean** mAcc	*figurines* mIoU	*figurines* mAcc	*teatime* mIoU	*teatime* mAcc	*ramen* mIoU	*ramen* mAcc	*waldo_kitchen* mIoU	*waldo_kitchen* mAcc
LangSplat [7]	3.28s	18GB	9.66	12.41	10.16	8.93	11.38	20.34	7.92	11.27	9.18	9.09
LEGaussians [8]	4.42s	5GB	16.21	23.82	17.99	23.21	19.27	27.12	15.79	26.76	11.78	18.18
OpenGaussian [11]	5.55s	9GB	**38.36**	51.43	39.29	55.36	**60.44**	76.27	**31.01**	**42.25**	22.70	31.82
SuperGSeg [ours]	**0.50s**	**4GB**	35.94	**52.02**	**43.68**	**60.71**	55.31	**77.97**	18.07	23.94	**26.71**	**45.45**

Table 2. Open-vocabulary 3D object selection comparison on the LERF-OVS dataset [7]. LERF [23] is not applicable for this task. We report the mIoU and mAcc of compared methods as provided in [11], and measure inference cost using their official implementations.

semantic segmentation (Section 4.3), it struggles in dense pixel-wise semantic segmentation. This is evident with occlusions due to projections onto 2D-pixel space. Without alpha blending, the occluded Gaussians cannot be effectively distinguished from one another. Instead, LangSplat [7] produces fine border segmentation but often includes incorrect semantic labels and noisy predictions, likely due to the lossy encoding of language information. LERF [23] presents accurate semantic prediction but with imprecise boundaries, limiting its applicability in fine-grained segmentation tasks.

4.3. Open-Vocabulary Object Selection

Quantitative Results. SuperGSeg improves over baseline methods that assign and optimize language features per Gaussian [7, 9, 16]. As shown in Table 2, clustering Gaussians into SuperGs enhances both spatial and semantic accuracy over per-Gaussian methods. We further compare SuperGSeg to OpenGaussian [11], another method exploring 3D Gaussian clustering. OpenGaussian's direct 2D CLIP feature association yields a slightly higher mIoU by avoiding alpha-blending artifacts, but it underperforms in 2D semantic segmentation on ScanNet (Section 4.2). In contrast, SuperGSeg maintains competitive mIoU for 3D object selection while surpassing OpenGaussian in 2D semantic segmentation, enhancing its versatility across real-world applications. Our higher mAcc, especially in complex LERF-OVS scenes such as *figurines* and *waldo kitchen*, reflects the precision of Super-Gaussian clustering and instance grouping. By accurately segmenting Gaussians in 3D, SuperGSeg renders more complete 2D masks with sharper boundaries, improving semantic consistency in challenging settings. In addition, SuperGSeg reduces inference latency to around 0.5s per query and decreases memory usage to 4GB, more than 50% lower than the next best baseline at 9GB. These improvements, enabled by SuperGs, demonstrate the potential for real-time querying on resource-constrained devices.

Qualitative Results. For visualization, we query language features in 3D space and render the resulting 3D masks to 2D. As shown in Figure 5, SuperGSeg delivers precise 3D object selection without spurious outliers and produces clearer boundaries. Thanks to the 3D understanding capability, our SuperGSeg allows for effective localization of occluded regions (e.g., the *stuffed bear* leg under a table). Notably, its high-quality features distinguish the coffee mug from its contents and spoon, showcasing the efficacy of distilling fine-grained features into SuperGs.

Ablation Study. We conduct ablation studies on various components of our method to validate the necessity of SuperGs, as summarized in Table 3. The baseline without SuperG (case a) trains the language feature field by directly optimizing per-anchor features, which results in limited semantic consistency. To analyze how different feature types affect SuperG formation, we evaluate grouping based solely on anchor coordinates and geometric features (case b), instance features (case c), and hierarchical features (case d). The results indicate that grouping Gaussians into SuperG improves semantic consistency compared to per-anchor optimization, but relying only on coordinates and geometry remains suboptimal. Both instance and hierarchical features contribute substantially to accurate SuperG assignments, and the best performance is achieved with our full model (case f), which combines both. We further compare K-means clustering for Gaussian grouping (case e) with our learnable SuperG assignment (case f). By dynamically adapting to variations in the feature space, our learnable predictor produces higher-quality SuperGs, yielding consistently higher mIoU and improved mAcc. Additional ablation studies on components of the SuperG clustering network are provided in Appendix D.

#	w/ Learned SuperG	w/ ins	w/ hier	mIoU ↑	mAcc. ↑
a)				10.12	14.49
b)	✓			12.08	16.95
c)	✓	✓		53.91	64.41
d)	✓		✓	49.04	66.10
e)		✓	✓	53.77	67.80
f)	✓	✓	✓	**55.31**	**77.97**

Table 3. SuperG ablation study, *teatime* scene of LERF-OVS.

4.4. Application

Beyond language-based querying, SuperGs serve as a multi-granularity representation of 3D scenes by integrating instance- and part-level knowledge, readily applicable to tasks such as cross-frame segmentation and hierarchical instance decomposition, without requiring task-specific retraining. For example, a click on a reference image retrieves SuperGs with matching hierarchical features, allowing the selected part to be consistently rendered across views. In addition to cross-view querying, SuperGSeg enables cross-level queries: clicking on a part retrieves its parent object using instance features, while clicking on an object reveals its constituent parts, which supports seamless navigation from parts to instances and vice versa, as illustrated in Figure 6. Furthermore, the granularity of instance-to-part segmentation can be adjusted by varying the threshold on hierarchical feature similarity, as shown in Figure 7. Additional implementation details are provided in Appendix A.

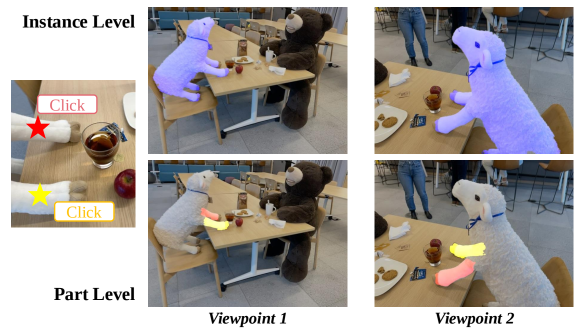

Figure 6. Cross-level and cross-frame segmentation visualization.

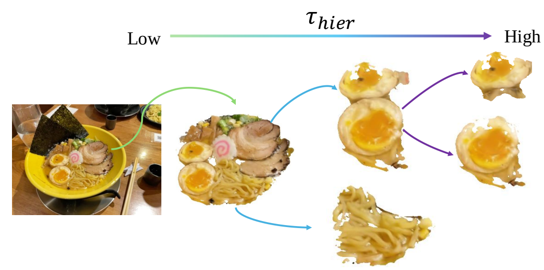

Figure 7. Visualization of intra-object hierarchy definition.

5. Conclusion

We present SuperGSeg, a novel framework for 3D scene understanding that represents scenes using compact Super-Gaussians, ensuring semantic and appearance consistency. By leveraging neural Gaussians, our method captures instance- and part-level segmentation features, guiding Super-Gaussian clustering through an adaptive online learning algorithm. Experiments show that integrating high-dimensional language features significantly improves open-set 3D language querying, demonstrating the framework's remarkable performance. Furthermore, the Super-Gaussian representation is readily adaptable to a wide range of 3D scene understanding tasks.

6. Acknowledgement

This work was supported by the German Research Foundation (DFG): SFB 1233, Robust Vision: Inference Principles and Neural Mechanisms, TP A1, project number: 276693517, and the Machine Learning Cluster of Excellence, EXC number 2064/1 – Project number 390727645. The authors thank the International Max Planck Research School for Intelligent Systems (IMPRS-IS) for supporting Siyun Liang.

References

[1] 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

[2] 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

[3] Zehao Yu, Torsten Sattler, and Andreas Geiger. Gaussian opacity fields: Efficient and compact surface reconstruction in unbounded scenes. *arXiv preprint arXiv:2404.10772*, 2024. 2

[4] Pinxuan Dai, Jiamin Xu, Wenxiang Xie, Xinguo Liu, Huamin Wang, and Weiwei Xu. High-quality surface reconstruction using gaussian surfels. In *ACM SIGGRAPH 2024 Conference Papers*, pages 1–11, 2024.

[5] Antoine Guédon and Vincent Lepetit. Sugar: Surface-aligned gaussian splatting for efficient 3d mesh reconstruction and high-quality mesh rendering. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 5354–5363, 2024. 2

[6] Taoran Yi, Jiemin Fang, Zanwei Zhou, Junjie Wang, Guanjun Wu, Lingxi Xie, Xiaopeng Zhang, Wenyu Liu, Xinggang Wang, and Qi Tian. Gaussiandreamerpro: Text to manipulable 3D gaussians with highly enhanced quality. *arXiv preprint arXiv:2406.18462*, 2024. 2

[7] 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. 2, 4, 5, 6, 7, 3

[8] Jin-Chuan Shi, Miao Wang, Hao-Bin Duan, and Shao-Hua Guan. Language embedded 3D gaussians for open-vocabulary scene understanding. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 5333–5343, 2024. 5, 6, 7, 4

[9] 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, 6, 7

[10] Mingqiao Ye, Martin Danelljan, Fisher Yu, and Lei Ke. Gaussian grouping: Segment and edit anything in 3D scenes. In *Proceedings of the European Conference on Computer Vision*, pages 162–179, 2024. 2

[11] 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. *arXiv preprint arXiv:2406.02058*, 2024. 2, 4, 6, 7, 3

[12] 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. *arXiv preprint arXiv:2411.19235*, 2024. 2, 5

[13] 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. 2, 3

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

[15] 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

[16] Yanyan Li, Chenyu Lyu, Yan Di, Guangyao Zhai, Gim Hee Lee, and Federico Tombari. Geogaussian: Geometry-aware gaussian splatting for scene rendering. In *European Conference on Computer Vision*, pages 441–457. Springer, 2024. 2, 6, 7

[17] Phuc Nguyen, Tuan Duc Ngo, Evangelos Kalogerakis, Chuang Gan, Anh Tran, Cuong Pham, and Khoi Nguyen. Open3Dis: Open-vocabulary 3D instance segmentation with 2d mask guidance. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 4018–4028, 2024. 2, 3

[18] Haiyang Ying, Yixuan Yin, Jinzhi Zhang, Fan Wang, Tao Yu, Ruqi Huang, and Lu Fang. Omniseg3D: Omniversal 3D segmentation via hierarchical contrastive learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 20612–20622, 2024. 2, 4

[19] Le Hui, Jia Yuan, Mingmei Cheng, Jin Xie, and Jian Yang. Superpoint network for point cloud oversegmentation. In *ICCV*, 2021. 2, 3, 5, 1

[20] 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, 6, 4

[21] Alexander Kirillov, Eric Mintun, Nikhila Ravi, Hanzi Mao, Chloe Rolland, Laura Gustafson, Tete Xiao, Spencer Whitehead, Alexander C Berg, Wan-Yen Lo, et al. Segment anything. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 4015–4026, 2023. 2, 3, 4

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

[23] 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. 2, 5, 6, 7, 4

[24] Francis Engelmann, Fabian Manhardt, Michael Niemeyer, Keisuke Tateno, Marc Pollefeys, and Federico Tombari. Opennerf: open set 3d neural scene segmentation with pixel-wise features and rendered novel views. *arXiv preprint arXiv:2404.03650*, 2024. 2

[25] Chung Min Kim, Mingxuan Wu, Justin Kerr, Ken Goldberg, Matthew Tancik, and Angjoo Kanazawa. Garfield: Group anything with radiance fields. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 21530–21539, 2024. 2, 4

[26] Loic Landrieu and Martin Simonovsky. Large-scale point cloud semantic segmentation with superpoint graphs. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 4558–4567, 2018. 3

[27] Damien Robert, Hugo Raguet, and Loic Landrieu. Efficient 3D semantic segmentation with superpoint transformer. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 17195–17204, 2023. 3

[28] Damien Robert, Hugo Raguet, and Loic Landrieu. Scalable 3D panoptic segmentation as superpoint graph clustering. In *2024 International Conference on 3D Vision (3DV)*, pages 179–189. IEEE, 2024.

[29] Loic Landrieu and Guillaume Obozinski. Cut pursuit: Fast algorithms to learn piecewise constant functions on general weighted graphs. *SIAM Journal on Imaging Sciences*, 10(4):1724–1766, 2017.

[30] Yun Zhu, Le Hui, Yaqi Shen, and Jin Xie. SPGroup3D: Superpoint grouping network for indoor 3D object detection. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 38, pages 7811–7819, 2024.

[31] Mingmei Cheng, Le Hui, Jin Xie, and Jian Yang. Sspc-net: Semi-supervised semantic 3D point cloud segmentation network. In *Proceedings of the AAAI conference on artificial intelligence*, volume 35, pages 1140–1147, 2021. 3

[32] Jeremie Papon, Alexey Abramov, Markus Schoeler, and Florentin Worgotter. Voxel cloud connectivity segmentation-supervoxels for point clouds. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 2027–2034, 2013. 3

[33] 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 conference on computer vision and pattern recognition*, pages 652–660, 2017. 3

[34] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. *Advances in neural information processing systems*, 30, 2017. 3

[35] Loic Landrieu and Mohamed Boussaha. Point cloud oversegmentation with graph-structured deep metric learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 7440–7449, 2019. 3

[36] Tao Lu, Mulin Yu, Linning Xu, Yuanbo Xiangli, Limin Wang, Dahua Lin, and Bo Dai. Scaffold-gs: Structured 3d gaussians for view-adaptive rendering. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 20654–20664, 2024. 3, 6, 2

[37] Matthias Zwicker, Hanspeter Pfister, Jeroen Van Baar, and Markus Gross. Ewa splatting. *IEEE Transactions on Visualization and Computer Graphics*, 8(3):223–238, 2002. 3

[38] 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. 4

[39] 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. *arXiv preprint arXiv:2405.17596*, 2024. 4

[40] Kim Jun-Seong, Kim GeonU, 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 *CVPR*, 2025. 4

[41] Myrna C Silva, Mahtab Dahaghin, Matteo Toso, and Alessio Del Bue. Contrastive gaussian clustering: Weakly supervised 3d scene segmentation. *arXiv preprint arXiv:2404.12784*, 2024. 4

[42] Charles Ruizhongtai Qi, Li Yi, Hao Su, and Leonidas J Guibas. Pointnet++: Deep hierarchical feature learning on point sets in a metric space. *Advances in neural information processing systems*, 30, 2017. 5

[43] Diwen Wan, Ruijie Lu, and Gang Zeng. Superpoint gaussian splatting for real-time high-fidelity dynamic scene reconstruction. In *Proceedings of the 41st International Conference on Machine Learning*, pages 49957–49972, 2024. 5

[44] Haoran Chen, Kenneth Blomqvist, Francesco Milano, and Roland Siegwart. Panoptic vision-language feature fields. *IEEE Robotics and Automation Letters*, 9(3):2144–2151, 2024. 6

[45] Diederik Kingma and Jimmy Ba. Adam: A method for stochastic optimization. In *International Conference on Learning Representations (ICLR)*, 2015. 2

[46] Sara Fridovich-Keil, Alex Yu, Matthew Tancik, Qinhong Chen, Benjamin Recht, and Angjoo Kanazawa. Plenoxels: Radiance fields without neural networks. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 5501–5510, 2022. 2