RETHINKING REPRESENTATION GRANULARITY IN OPEN-VOCABULARY 3DGS: ENDOGENOUS SEMANTICS VIA SPARSE ANCHORS

Anonymous authors
Paper under double-blind review

ABSTRACT

Open-vocabulary semantic understanding with 3D Gaussian Splatting (3DGS) has long been constrained by two untested representation premises, namely that semantics rely on dense per-primitive representations over the full set of primitives, and that semantics must take the form of independent extrinsic high-dimensional parameters, rather than being intrinsic properties of geometric primitives. From the perspective of representation learning, this paper systematically revisits representation granularity and semantic sources using structured sparse anchors as an experimental platform. We propose Endogenous Semantic Mapping: it discards traditional extrinsically attached semantic vectors, takes only the intrinsic geometric and appearance features of sparse anchors as input, and directly generates semantics through a globally shared network. Specifically, semantic vectors are embedded as native rendering attributes of 3D Gaussians, and pixel-wise semantic features are directly output through the native rasterization pipeline, rather than being predicted by an external network and attached as auxiliary properties. This method achieves zero scene-level semantic parameter storage and can directly reuse pre-trained geometric models. Controlled experiments on multiple public benchmarks reveal a key finding that dense Gaussian representations exhibit significant over-parameterization for semantic tasks. When semantic representation units are compressed to approximately 10% of the full dense primitives, no significant degradation in segmentation accuracy is observed, demonstrating a fundamental mismatch between the optimal representation granularities required for rendering and semantics. At this extremely sparse granularity, the proposed method not only significantly reduces storage and VRAM overhead, but also achieves competitive segmentation accuracy against mainstream methods that adopt full dense representations with extrinsically attached semantics on public 3D semantic benchmarks. This study empirically confirms that native structured scene features possess strong semantic encoding capacity, and establishes a new design paradigm for lightweight 3D semantic representation.

Keywords: 3D Gaussian Splatting (3DGS); open-vocabulary 3D semantics; representation redundancy; structured sparse anchors; endogenous semantic mapping

1 INTRODUCTION

Three-Dimensional Gaussian Splatting (3DGS) (Kerbl et al., 2023) enables high-fidelity real-time rendering and has become a mainstream technique for 3D scene reconstruction and understanding. Yet mainstream open-vocabulary semantic methods hold to two untested premises at the representation level. They take the dense, per-Gaussian representation over all primitives as the basis for semantic reasoning, assuming a denser representation always yields more accurate semantics, and they treat semantics as independent parameters that must be additionally attached, assuming semantics cannot arise from native geometric and appearance features. This raises two questions, namely how dense a representation semantic understanding actually requires, and whether semantics must be attached at all. If a small number of structured representations already suffice to support high-quality semantics, carrying semantics over the full set of dense Gaussians is over-parameterized, and the attached parameters may not be necessary.

Reviewers: please read the Reviewer Guidelines (iclr.cc/Conferences/2027/ReviewerGuidelines) and the AI Policy for Reviewers (iclr.cc/Conferences/2027/AIPolicyForReviewers). If you used AI to expand, edit, or polish your review, please provide the input text to the LLM. Better still, consider skipping the LLM and submitting your original text: we, and the authors, are much more interested in your unedited thoughts than in what an LLM has to say. AI-assisted or not, you are putting your name and reputation behind your review: LLM-generated falsehoods, hallucinations or misrepresentations are subject to disciplinary action, which may include desk-rejecting all papers you have authored.

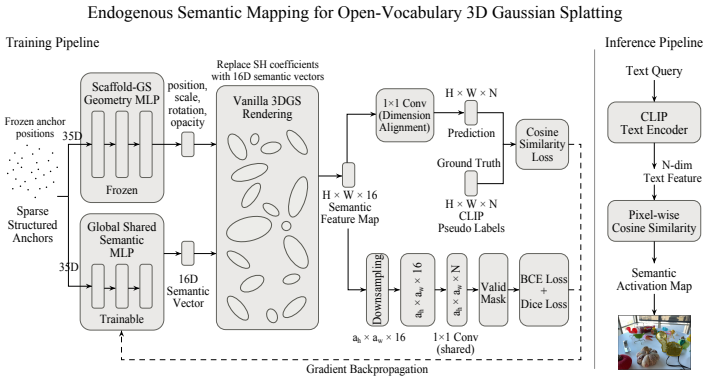


Figure 1: Overall framework. Sparse anchors of Scaffold-GS are the only input, and a single shared network maps their native geometric and appearance features to the semantic space without adding semantic parameters; rendered features are then lifted to the CLIP text space.

Existing attempts to improve semantic efficiency mostly remain within the premise of full representations and fall into two types of post-processing. Semantics-guided geometric pruning first obtains the full set of Gaussians and complete semantics and then removes some units, an after-the-fact dense-to-sparse compression whose excessive sparsity leads to clear accuracy loss, while hierarchical semantic indexing still retains all Gaussians at the bottom and only clusters or externalizes features at an upper level, so the underlying storage stays on par with the full scheme. Some works also bypass the attached semantic field but remain extrinsic. Semantic Gaussians (Guo et al., 2026) regresses per-primitive embeddings from native Gaussian attributes through a 3D network, yet it voxelizes the Gaussians and applies 3D convolution for inference, bypassing rasterization, with tens of millions of parameters while still storing an embedding for each Gaussian. Segment Then Splat (Lu et al., 2025) removes per-primitive vectors and obtains instance-level groups from trained masks to reduce storage, but the semantics still come from the forward results of 2D models, so the extrinsic nature is unchanged.

Attached semantics also presuppose that geometry and semantics are independent of each other, so semantics can only be aligned point-wise through back-projection of 2D features, incurring storage and computation that grow with scene scale while denying the semantic encoding capacity of native features. Whether a learnable endogenous mapping exists among geometry, appearance, and semantic categories has so far lacked quantitative examination. To examine semantic redundancy cleanly, we need a platform that substantially reduces the number of representations while fully preserving the core structure, since directly down-sampling the full set of Gaussians would damage the geometry and the change in accuracy could then not be attributed to redundancy. The structured anchors of Scaffold-GS (Lu et al., 2024) meet this need precisely. It generates sparse anchors at key geometric positions, and the anchors derive child Gaussians for rendering; the total representation volume is usually about 10% of the full set of Gaussians, while core geometric topology and appearance are preserved. Anchors are independent optimization and representation units, and the number of child Gaussians they derive is adjustable, providing a controllable platform for quantitatively studying semantic granularity.

On this basis, we further argue that semantics are the projection of native representations into the semantic space rather than independent extrinsic parameters. The local shape, curvature, scale, and appearance statistics encoded by anchors correspond stably to semantic concepts and can emerge endogenously through a shared mapping, without per-primitive storage. We take the native features of sparse anchors as the only input for semantic reasoning and learn a globally shared mapping that produces no scene-level semantic storage throughout. The method is built on a pre-trained Scaffold-GS model, freezing and reusing all of its scene representations without retraining the reconstruction.

The main contributions of this paper are as follows.

1. We systematically examine how open-vocabulary 3D semantic performance varies with representation density and provide empirical evidence on the order of representations required for semantic tasks, showing that dense representations optimized for rendering are

clearly over-parameterized for semantics and that semantic information is concentrated in a small number of structured core units.

2. We propose an endogenous semantic mapping paradigm built on sparse anchors. Without attaching any independent semantic vectors, codebooks, or external indices, it achieves zero scene-level semantic storage, and using only about 10% of the representation units, it surpasses mainstream methods with full dense representations and attached semantics on the indoor benchmark.

2 Related Work

2.1 Per-Gaussian Semantic Embedding Paradigm

Independently attaching a semantic vector to each Gaussian is the foundational paradigm of the field, the core of which is to align high-dimensional features with per-Gaussian representations through 2D supervision. LangSplat (Qin et al., 2024) first distills CLIP embeddings into 3D Gaussians and establishes this framework. OpenGaussian (Wu et al., 2024) achieves point-level understanding through instance-level 3D-to-2D association and a two-stage codebook, LaGa (Cen et al., 2025b) alleviates view dependence with multi-view aggregation, CCL-LGS (Tian et al., 2025) improves inter-class discrimination with a contrastive codebook, Splat Feature Solver (Xiong et al., 2026) casts feature solving as a regularized sparse linear inverse problem, and GenSplat (Liu et al., 2026a) further alleviates the distribution mismatch between 2D features and 3D primitives. Other works have different focuses, addressing multi-view inconsistency (LEGaussians (Shi et al., 2024), EconSG (Zhang & Lee, 2025)), SAM-feature distillation (Feature3DGS (Zhou et al., 2024)), vision-language alignment (3D-VLGS (Peng et al., 2025)), cross-view context fusion (ProFuse (Chiou et al., 2026)), and granularity-aware distillation (GAGS (Peng et al., 2026)). Semantic Gaussians (Guo et al., 2026) predicts per-primitive embeddings through an independent 3D network, Segment Then Splat (Lu et al., 2025) avoids per-pixel ambiguity through segmentation before reconstruction, LangFlash (Liu et al., 2026b) and SparseLGS (Hu et al., 2024) reduce per-scene optimization cost through feed-forward generation and sparse views, and SA3D (Cen et al., 2023) and Semantic-Consistent Reconstruction (Zhang et al., 2025) directly generate segmentation relying on SAM cross-view mask propagation. This thread always takes the full dense Gaussians as the substrate, neither examining their semantic redundancy nor exploring endogenous semantics.

2.2 Hierarchical and Instance-Level Semantic Organization

To address the high storage overhead of per-Gaussian semantics and weak instance consistency, one line of work raises the granularity of semantic attachment to instances or hierarchical groups. Gaussian Grouping (Ye et al., 2024) introduces a compact identity encoding for each Gaussian and obtains object-level grouping through SAM masks and cross-view association. InstanceGaussian (Li et al., 2025) lets the child Gaussians derived from an anchor share the same semantic vector. ReLaGS (Xie et al., 2026) introduces relation extraction and a 3D scene graph, CausalSplat (Ding et al., 2026b) improves relation inference with causal constraints, ReasonGrounder (Liu et al., 2025) adaptively groups according to physical scale, and OpenSplat3D (Piekenbrinck et al., 2025) combines instance masks and language embeddings. Click-Gaussian (Choi et al., 2024), SAGA (Cen et al., 2025a), SuperGSeg (Liang et al., 2026), ReferSplat (He et al., 2025), THGS (Dai et al., 2025), and OnlinePG (Zhai et al., 2026) extend semantic organization from interactive segmentation, multi-granularity affinity, super-Gaussian hierarchies, referring segmentation, Transformer hierarchies, and online mapping, respectively. Such methods optimize the granularity of semantic organization rather than the order of representations, and most still start from the full set of Gaussians, with semantics attached as independent fields or external indices (Ding et al., 2026a). Taking InstanceGaussian (Li et al., 2025) as an example, its semantic vectors are still attached per anchor and storage grows with the number of anchors, while semantic learning must proceed synchronously with reconstruction, so it cannot freeze an existing reconstruction and add semantics separately. We instead replace all semantic vectors with a lightweight MLP shared across space, with semantic-side parameters independent of the number of anchors, and remove per-primitive fields on a frozen platform, differing essentially from such methods.

2.3 Semantic Lightweighting and Geometric Post-Processing Pruning

Existing post-processing-style lightweighting proceeds along semantic-level compression and geometric-level pruning. At the semantic level, ExtrinSplat (Ding et al., 2026a) replaces per-Gaussian high-dimensional features with decoupled external text indices, and CSGaussian (Tseng et al., 2026) progressively compresses from a rate-distortion perspective while accommodating segmentation, but the attached volume is still determined by scene structure. At the geometric level, pruning often serves as front-end denoising. The maximum weight pruning of ReLaGS (Xie et al., 2026) removes low-contribution points according to the maximum contribution of a Gaussian across views, with a limited ratio that does not change the foundational status of full representations. Two types of redundancy should be distinguished. Geometric pruning targets visual redundancy with rendering quality as the yardstick, and after pruning it is still a dense representation serving rendering. We instead examine semantic redundancy, namely the marginal contribution of units to the semantic task. Many Gaussians that reproduce smooth appearance are necessary for rendering yet provide almost no incremental class discrimination, so this cannot be verified through geometric pruning. Even when anchor-level context modeling (Wang et al., 2024) and rate-distortion compression (Tseng et al., 2026) can substantially reduce primitives, the optimization objective is always reconstruction fidelity. The two underlying questions of what density semantics require and whether semantics must be extrinsic remain unanswered, and this paper is conducted precisely around them.

3 Method

The overall framework of the proposed method is shown in Figure 1. We take the structured anchors of Scaffold-GS as the only representation carrier and construct anchors encoding native geometry and appearance from a sparse point cloud. We then build a globally shared lightweight mapping network over the anchors, mapping native features into a low-dimensional semantic latent space, and obtain pixel-level semantic features through differentiable rendering and dimension lifting with 1×1 convolutions. During training, supervision comes from semantic pseudo-labels generated by 2D foundation models. During inference, open-vocabulary queries are answered through the similarity between CLIP text features and the rendered semantic features.

3.1 Structured Anchor Representation of Scaffold-GS

Scaffold-GS (Lu et al., 2024) starts from the sparse point cloud obtained by structure-from-motion and constructs a sparse anchor grid through voxelization. Each anchor carries a feature vector and scale and serves as an independent optimization and storage unit. During rendering, for each visible anchor, a decoding network combines its feature vector with view information to predict the center offsets, opacities, colors, and covariances of a number of child Gaussians. The child Gaussians themselves have no independent parameters and are only intermediate rendering primitives. Compared with the full dense Gaussians of the original 3DGS, this structure substantially compresses the total representation volume while preserving core geometric topology and appearance details.

Scaffold-GS is chosen as the examination platform of this paper for three reasons. The first is native sparsity. Sparsity is endogenous to the representation system, rather than being applied as after-the-fact pruning to a trained dense model. In the test scenes, the number of anchors is only about one-tenth that of the full dense Gaussians, allowing a direct, like-for-like comparison with the mainstream full-dense paradigm. The second is that the direction of sparsity is consistent with the direction of semantic redundancy. The distribution of anchors, driven by rendering losses, tends to concentrate in regions with significant geometric variation, while very few units are allocated to smooth-appearance regions; these primitives are necessary for reproducing smooth appearance yet contribute the least to discriminating semantic classes, so if semantic performance at the anchor level does not degrade noticeably, it can be directly attributed to the existence of semantic redundancy rather than damage to the spatial structure. The third is adjustable density. Anchor density can be adjusted through voxel size and growth/pruning strategies, making it easy to characterize the relationship between density and performance. Once the geometry is frozen, the anchor features become fixed low-dimensional scene descriptors, providing the basis for constructing the semantic mapping at the anchor level and removing attached fields. All anchors in the scene share the same set of network weights, and only a lightweight MLP projects the anchor features into the semantic space, as detailed in Section 3.2.

3.2 Endogenous Semantic Mapping from Anchor Features

The basic assumption of this paper regarding the source of semantics is that semantics are endogenous to the structural and appearance features of the scene. The geometric topology, spatial layout, and appearance texture of a 3D scene are inherently the physical carriers of semantic concepts, so there is no need to independently store semantic parameters for each spatial point. Instead, one can learn a mapping from native features to the semantic space that generates semantic expressions when needed. Following this assumption, this paper builds the endogenous semantic mapping pipeline using anchors as the basic units, without introducing any scene-bound independent semantic vectors, and all semantic information is obtained from the native anchor features through the shared mapping.

We formalize the input of each anchor as a 35-dimensional joint vector. Its main body is the native 32-dimensional anchor structural feature $\boldsymbol{f}_i \in \mathbb{R}^{32}$, which already encodes the geometric shape, curvature, scale, and appearance statistics of the local region during reconstruction, supplemented by the 3-dimensional spatial coordinate $\boldsymbol{p}_i \in \mathbb{R}^3$, which provides a global position constraint to distinguish semantic entities of the same shape at different locations. Concatenating the two yields the joint vector $\boldsymbol{x}_i = [\boldsymbol{f}_i^\top, \boldsymbol{p}_i^\top]^\top \in \mathbb{R}^{35}$.

The semantic mapping is implemented by a globally shared lightweight MLP containing four fully connected layers, each followed by a ReLU activation, which nonlinearly maps the 35-dimensional joint feature into a 16-dimensional semantic latent space, obtaining the anchor semantic vector $\boldsymbol{z}_i = \Phi_\theta(\boldsymbol{x}_i) \in \mathbb{R}^{16}$. Because the network parameters are globally shared and not replicated across scenes or anchors, the scene side stores no semantic vectors and incurs no scene-growing semantic storage. Semantic knowledge is encoded in the shared mapping function rather than attached per primitive, and the scene data still contains only geometric and appearance representations, achieving zero additional semantic storage.

After obtaining the anchor semantic vectors, we reuse the child-Gaussian generation logic of Scaffold-GS and the native differentiable rendering pipeline of 3DGS for the spatial projection of semantics. The 16-dimensional semantic vector replaces the spherical-harmonic color features of the original pipeline and, together with the derived child Gaussians, participates in view projection, rasterization sorting, and alpha blending. Denoting the rendering operator as $\mathcal{R}$, it outputs a 16-channel semantic feature map of resolution $H \times W$ for the corresponding view:

$$\boldsymbol{S} = \mathcal{R}(\{\boldsymbol{z}_i\}) \in \mathbb{R}^{16 \times H \times W}. \tag{1}$$

This design fully reuses the original geometric alignment logic and requires no additional 3D-to-2D projection operator, ensuring precise alignment between semantics and geometry while retaining the efficiency of the original pipeline to the greatest extent. The semantic map shares the same rasterization and alpha-blending pipeline as the color map, and the loss is computed on the pixel-level semantic map and back-propagated end-to-end through the renderer, without an independent semantic prediction network.

To align with the open-vocabulary supervision space, we construct a dimension alignment module Ψ from two 1×1 convolutions, which progressively lifts the 16-channel semantic map into a D-dimensional space consistent with OpenCLIP text features, producing the pixel-level semantic features $\hat{\boldsymbol{Y}} = \Psi(\boldsymbol{S}) \in \mathbb{R}^{D \times H \times W}$ for supervised training. The inference stage supports open-vocabulary queries and shares the complete pipeline with training, without retraining or fine-tuning. We adopt the standard query protocol consistent with LangSplat (Qin et al., 2024) to ensure fair comparison. Any text is encoded by the CLIP text encoder, and a per-pixel cosine similarity is computed against the rendered and lifted pixel features, followed by thresholding and standard post-processing to produce the segmentation. The method also supports an engineering acceleration of anchor-level semantic pre-filtering, which can improve inference efficiency on large scenes with negligible accuracy loss.

3.3 Training Strategy and Loss Function

3.3.1 Training Paradigm and Supervision Data

The entire semantic mapping pipeline is trained end-to-end. During training, all scene representation parameters of Scaffold-GS, including anchor positions, structural features, child-Gaussian offsets, and appearance attributes, are completely frozen, and only the semantic mapping network Φ_θ and the dimension alignment convolutions Ψ are optimized. This paradigm keeps the scene geometry intact while making training computationally light and fast to converge.

Training uses offline, pre-generated pixel-level semantic pseudo-labels as supervision, following the common protocol of open-vocabulary 3D semantic tasks (Qin et al., 2024). For the 2D image of a training view, a large-scale vision-language model first performs initial semantic parsing to obtain candidate regions and category annotations. The candidate regions are then fed into SAM (Kirillov et al., 2023) to produce pixel-level, multi-granularity masks. Finally, OpenCLIP is used to extract and fuse image and text embeddings, constructing a D-dimensional pixel semantic feature map aligned with OpenCLIP as the supervision target, with which the differentiably rendered pixel features are aligned and back-propagated. These 2D models provide offline supervision only during training and are completely removed at inference.

3.3.2 Loss Function Design

The loss is a weighted combination of the basic semantic embedding loss and the Semantic False-Negative Suppression (SFNS) enhancement loss:

$$\mathcal{L} = \lambda_e \mathcal{L}_{\text{embed}} + \lambda_s \mathcal{L}_{\text{SFNS}}, \tag{2}$$

where λ_e and λ_s are the two weights. The basic semantic embedding loss adopts a cosine similarity loss that constrains the global alignment in vector space between the rendered pixel features and the pseudo-labels, serving as the core supervision signal:

$$\mathcal{L}_{\text{embed}} = 1 - \frac{1}{H \cdot W} \sum_{h=1}^{H} \sum_{w=1}^{W} \frac{\hat{\boldsymbol{y}}_{h,w} \cdot \boldsymbol{y}^{\text{gt}}_{h,w}}{\|\hat{\boldsymbol{y}}_{h,w}\| \cdot \|\boldsymbol{y}^{\text{gt}}_{h,w}\|}, \tag{3}$$

where $\hat{\boldsymbol{y}}_{h,w}$ is the pixel feature output by the model and $\boldsymbol{y}^{\text{gt}}_{h,w}$ is the corresponding pseudo-label.

To address the blurred boundaries between semantically similar classes and the false-negative penalties induced by pseudo-labels, we further refine accuracy with the combined SFNS enhancement loss, which consists of pixel-level BCE and Dice losses:

$$\mathcal{L}_{\text{SFNS}} = w_b \mathcal{L}_{\text{BCE}} + w_d \mathcal{L}_{\text{Dice}}. \tag{4}$$

The BCE provides per-pixel classification supervision to ensure basic accuracy, while the Dice alleviates the foreground-background imbalance and improves region overlap and boundary clarity. Together they suppress the false-negative effects of pseudo-label noise and enhance boundary quality. Training proceeds in two stages. The first stage uses only the basic embedding loss for the main training, completing the global alignment of pixel features with the CLIP space and establishing the basic semantic response. The second stage introduces the combined SFNS loss for fine-tuning on top of the first-stage weights, refining blurred boundaries and small-object segmentation, and is especially suited to indoor scenes with stacked objects and complex occlusion.

BCE and Dice are standard optimization items in semantic segmentation, and comparable methods such as CCL-LGS (Tian et al., 2025) and ReferSplat (He et al., 2025) also adopt equivalent pixel-level segmentation losses to refine boundaries. The performance advantage of this paper lies primarily in the semantic efficiency of structured sparse representations and the endogenous mapping paradigm. While second-stage refinement is necessary for strong performance, our contribution is orthogonal to such optimization, lying in how semantics are represented rather than refined.

4 EXPERIMENTS

In this chapter, we first quantitatively characterize how semantic performance varies with representation density through controlled experiments, examining the semantic redundancy of dense Gaussian representations. We then validate open-vocabulary segmentation performance on a standard benchmark, and finally report the storage and GPU memory overhead.

4.1 DATASETS, EVALUATION METRICS, AND IMPLEMENTATION DETAILS

Datasets. We select LERF-OVS (Kerr et al., 2023) as the core evaluation benchmark. It is built on real captured complex indoor scenes, consisting of everyday environments with multiple stacked objects, occlusion, and rich textures, and covering multi-level open-vocabulary annotations of objects, parts, and materials. In addition, our validation results on ScanNet (Dai et al., 2017) are provided in Appendix A.1.

Evaluation metrics. We use the mean Intersection over Union (mIoU) to measure the spatial overlap between segmentation results and ground-truth regions, with higher values indicating higher accuracy. Query matching and post-processing strictly follow the standard setup of LangSplat (Qin et al., 2024).

Implementation details. The method is built on a pre-trained Scaffold-GS geometric model. During semantic mapping training, all geometric parameters are frozen and only the semantic mapping modules are optimized. The semantic mapping network is a four-layer MLP (widths $35\to128\to1024\to128\to16$, with ReLU after each layer), with about 270K parameters, which maps the 35-dimensional anchor joint feature into the 16-dimensional semantic latent space. It is then followed by two 1×1 convolutions that lift the dimension to align with the OpenCLIP text space. The OpenCLIP feature dimension used in our experiments is $D=768$. The total trainable parameters of the semantic branch amount to about 371K; they are globally shared and do not grow with the number of scenes or anchors.

Training adopts a two-stage strategy using the Adam optimizer with an initial learning rate of 5×10^{-4}. The first stage trains for 10,000 iterations using only the basic embedding loss to complete global alignment, and the second stage fine-tunes for 4,000 iterations by introducing the combined SFNS loss to refine boundaries. The two weights are $\lambda_e=1.0$ and $\lambda_s=0.5$. All experiments are conducted on a single NVIDIA RTX 4090.

Baselines. We select formally published works from multiple technical routes as baselines, covering per-Gaussian embedding, multi-view aggregation, inverse-rendering solving, and contrastive learning, and including works from CVPR, ICCV, ICML, and ICLR. All baseline data are taken from the publicly reported results of the original papers on the same dataset.

4.2 REPRESENTATION EFFICIENCY AND REDUNDANCY ANALYSIS

In this section, controlled experiments quantitatively characterize the relationship between semantic representation density and segmentation accuracy, directly verifying the semantic redundancy in dense per-Gaussian representations. All experiments are conducted on LERF-OVS.

4.2.1 SPARSITY SENSITIVITY ANALYSIS

By setting different anchor generation parameters k, we analyze the effect of changes in the proportion of semantic representation units on mIoU. We set k to 2, 5, 10, and 20. The anchor count is governed by scene geometry and varies only modestly across k, while rendered Gaussians grow linearly with k. Increasing k multiplies Gaussians per anchor without proportionally adding semantic units, lowering semantic density. Smaller k places this density closer to the dense Gaussian representation.

As shown in Figure 2, the curve exhibits a clear flat plateau. Across $k = 2$, 5, 10, and 20, mIoU stays within a narrow band of 67.6 to 69.8 percent, with no systematic trend as sparsity varies. At the operating point $k = 10$, our method already outperforms all baselines in Table 1, surpassing mainstream schemes using the full dense representation with about one-tenth of the representation volume. When sparsity is pushed to 5%, i.e. $k = 20$, mIoU holds at 69.0%, still higher than all per-Gaussian baselines in Table 1, indicating that dense per-Gaussian representations contain substantial semantic redundancy.

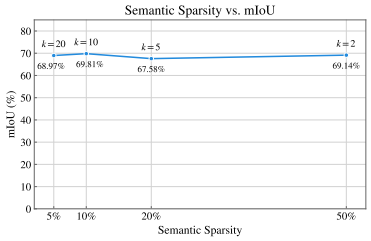


Figure 2: mIoU under different proportions of semantic units. The horizontal axis is the ratio of anchors to dense Gaussians and the vertical axis is mIoU, for $k = 2, 5, 10$, and 20.

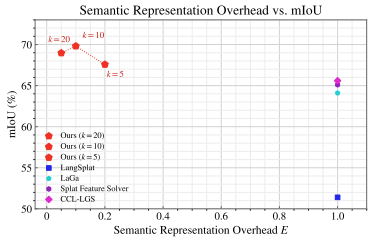


Figure 3: Semantic representation overhead. The horizontal axis is the overhead ratio E and the vertical axis is mIoU. Per-Gaussian methods cluster around $E{\approx}1.0$, whereas ours lies at $E{\approx}0.1$ with the highest accuracy.

This plateau has a clear representational meaning. Anchors concentrate at object boundaries and structural turning points, where classes switch and semantic information is densest, while many rendering-oriented Gaussians only reproduce smooth surfaces and gradual appearance and contribute little to semantic discrimination. Along the plateau, the removed units are mainly these low-contribution ones, so accuracy holds and declines markedly only after structured anchors are lost. The granularity required by rendering is thus inconsistent with that required by semantics. The dense representation best for rendering is over-parameterized for semantics, and semantic information concentrates on a small number of structured key points.

4.2.2 Semantic Representation Overhead Comparison

To quantify the relative scale of the semantic branch, we define the overhead ratio $E = N_{\text{sem}}/N_{\text{geo}}$, where N_{sem} is the number of independent units supporting semantic reasoning and matching and N_{geo} is the total number of rendered Gaussians; smaller values mean lower overhead. In bottom-up hierarchical methods, semantic units still aggregate the full set of Gaussians as an intermediate substrate, whereas ours are built directly on sparse anchors, achieving lightweighting at the root. Projecting our method and the per-Gaussian baselines onto the (E, mIoU) plane gives Figure 3.

Per-Gaussian methods represented by LangSplat independently maintain a semantic representation for each rendered Gaussian, with E around 1.0 and mIoU distributed from 51% to 66%. Our E is around 0.1 with mIoU reaching 69.8%, located in the efficient upper-left region. We achieve higher accuracy using only about one-tenth the semantic scale of per-Gaussian methods, and this efficiency comes from the native sparse representation and endogenous mapping rather than secondary compression after dense representations.

4.3 Open-Vocabulary Semantic Segmentation Results

We validate open-vocabulary segmentation accuracy in indoor environments on LERF-OVS, and the comparison results are shown in Table 1.

Table 1: Open-vocabulary semantic segmentation results on the LERF-OVS dataset (mIoU %).

Method	Figurines	Teatime	Ramen	Kitchen	Mean
LangSplat (Qin et al., 2024) (CVPR 2024)	44.7	65.1	51.2	44.5	51.4
InstanceGaussian (Li et al., 2025) (CVPR 2025)	45.5	63.4	24.6	29.2	40.7
ExtrinSplat (Ding et al., 2026a) (CVPR 2026)	66.4	64.4	45.6	40.9	54.3
ReferSplat (He et al., 2025) (ICML 2025)	67.5	50.1	55.1	48.9	55.4
LaGa (Cen et al., 2025b) (ICML 2025)	64.1	70.9	55.6	<u>65.6</u>	64.1
ReLaGS (Xie et al., 2026) (CVPR 2026)	64.7	**81.0**	51.2	60.6	64.4
Splat Feature Solver (Xiong et al., 2026) (ICLR 2026)	<u>67.6</u>	68.5	<u>62.3</u>	62.1	65.1
CCL-LGS (Tian et al., 2025) (ICCV 2025)	61.2	71.8	<u>62.3</u>	**67.1**	<u>65.6</u>
3D-VLGS (Peng et al., 2025) (ICLR 2025)	58.1	73.5	61.4	54.8	62.0
Ours (k=10)	**70.5**	<u>77.1</u>	**68.0**	63.7	**69.8**

Using only sparse anchors and storing no semantic features in the scene, our method exceeds the listed baselines with an average mIoU of 69.8%. The shape, curvature, and appearance statistics encoded by anchors already support semantic generation, confirming the feasibility of the endogenous mapping. The BCE and Dice boundary terms are standard segmentation items, so the gain does not

come from additional losses, and competitiveness is retained while both representation scale and semantic storage are sharply reduced.

To visualize the segmentation quality, Figure 4 presents the qualitative results of our method on LERF-OVS.

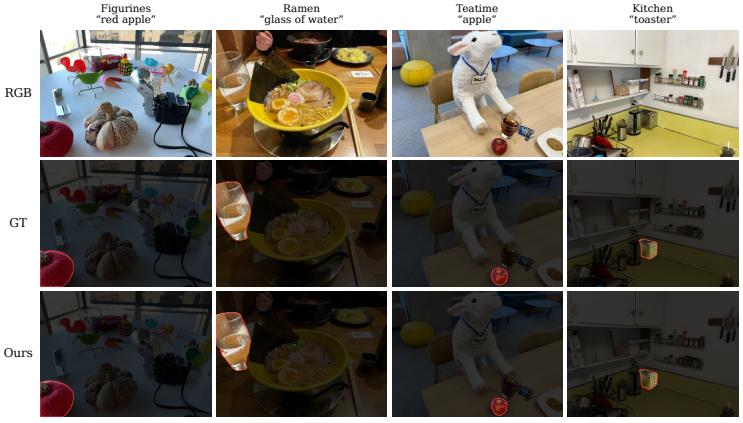


Figure 4: Qualitative results on LERF-OVS for the text queries shown above each column. Rows show RGB images, ground truth (GT), and our predictions.

4.4 Storage and GPU Memory Overhead

Mainstream methods store independent semantic vectors on primitives and load them with the scene. Even lightweighting schemes only compress or externalize these features without leaving the attached architecture. In our zero in-scene storage paradigm, semantics are generated at inference from anchor features through the shared mapping, and the scene holds no semantic vectors.

We benchmark hardware efficiency on the LERF Figurines scene, with baseline data cited from ExtrinSplat (Ding et al., 2026a). Results are shown in Table 2.

Table 2: Comparison of storage and GPU memory overhead across methods (LERF Figurines scene).

Method	Peak GPU memory	Per-scene semantic storage	mIoU
LangSplat (Qin et al., 2024) (CVPR 2024)	20 GB	~3 GB	44.7
OpenGaussian (Wu et al., 2024) (NeurIPS 2024)	22 GB	~3 GB	39.3
LaGa (Cen et al., 2025b) (ICML 2025)	24 GB	~3 GB	64.1
ExtrinSplat (Ding et al., 2026a) (CVPR 2026)	8 GB	~3 MB	66.4
3D-VLGS (Peng et al., 2025) (ICLR 2025)	24 GB	~3 GB	58.1
InstanceGaussian (Li et al., 2025) (CVPR 2025)	24 GB	~3 GB	45.5
Ours (k=10)	**7.09 GB**	**0**	**70.5**

Traditional per-Gaussian methods require about 3 GB of per-scene semantic storage and over 20 GB of peak GPU memory, whereas ExtrinSplat reduces storage to the MB order at about 8 GB. Our measured peak memory is 7.09 GiB on this scene, with no semantic storage beyond the globally shared weights, while accuracy stays competitive.

5 Conclusion

This paper studies open-vocabulary 3D semantics through the lens of Scaffold-GS anchors. We show that dense Gaussians are semantically redundant and that semantics can be generated from the geometric and appearance features of anchors by a shared mapping, with no semantic parameters stored in the scene. Dense representations optimized for rendering are over-parameterized for semantics, the granularities required by the two tasks differ, and semantic information concentrates on a small number of structured anchors. With about one-tenth of the units and zero in-scene storage, our method matches or exceeds full-dense methods with attached semantics at a fraction of the storage and GPU memory, showing that native structured features already possess strong semantic encoding capacity.

AI USE STATEMENT

We used AI-assisted tools during the writing and production of this paper. Language polishing, translation, and bibliography organization were assisted by a large language model, while the core academic ideas, reasoning, experimental design, and result analysis were completed independently by the authors. Figures and tables were first sketched and laid out by the authors, after which AI tools assisted with color schemes and initial drafts, with the authors verifying the final versions. All AI-assisted content was reviewed and confirmed by the authors, who take full responsibility for the paper's content.

ETHICS STATEMENT

This work does not involve human subjects, animal experiments, or personal privacy data. The LERF-OVS and ScanNet datasets used in the experiments are both publicly available benchmarks in the field and are used in compliance with their release agreements and terms of use.

REPRODUCIBILITY STATEMENT

To ensure reproducibility of the core results, we will provide the inference code, pre-trained semantic mapping MLP weights, and example scene anchor models in the supplementary material, allowing full replication of the segmentation, memory tests, and qualitative results. The complete training code and model files are large and will be open-sourced after the paper is accepted.

REFERENCES

J. Cen, Z. Zhou, J. Fang, C. Yang, W. Shen, L. Xie, D. Jiang, X. Zhang, and Q. Tian. Segment Anything in 3D with NeRFs. In *Advances in Neural Information Processing Systems (NeurIPS)*, pp. 25971–25990, 2023.

J. Cen, J. Fang, C. Yang, et al. Segment Any 3D Gaussians. In *Proceedings of the AAAI Conference on Artificial Intelligence (AAAI)*, 2025a.

J. Cen, X. Zhou, J. Fang, C. Wen, L. Xie, X. Zhang, W. Shen, and Q. Tian. Tackling View-Dependent Semantics in 3D Language Gaussian Splatting. In *International Conference on Machine Learning (ICML)*, 2025b.

Y.-J. Chiou, W.-T. Cheng, and Y.-F. Yang. ProFuse: Efficient Cross-View Context Fusion for Open-Vocabulary 3D Gaussian Splatting. arXiv:2601.04754, 2026.

S. Choi, H. Song, J. Kim, et al. Click-Gaussian: Interactive Segmentation to Any 3D Gaussians. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pp. 289–305, 2024.

A. Dai, A. X. Chang, M. Savva, M. Halber, T. Funkhouser, and M. Nießner. ScanNet: Richly-Annotated 3D Reconstructions of Indoor Scenes. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 5828–5839, 2017.

S. Dai, Y. Qu, Z. Li, X. Li, S. Zhang, and L. Cao. Training-Free Hierarchical Scene Understanding for Gaussian Splatting with Superpoint Graphs. In *Proceedings of the 33rd ACM International Conference on Multimedia (ACM MM)*, pp. 3673–3682, 2025.

J. Ding, X. Liu, Z. Pan, S. Long, and G. Li. ExtrinSplat: Decoupling Geometry and Semantics for Open-Vocabulary Understanding in 3D Gaussian Splatting. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2026a.

J. Ding, M. Song, Y. Chen, W. Gao, and G. Li. CausalSplat: Towards Comprehensive Hierarchical Reasoning in 3D Gaussian Splatting. In *Proceedings of the ACM International Conference on Multimedia (ACM MM)*, 2026b.

J. Guo, X. Ma, Y. Fan, H. Liu, and Q. Li. Semantic Gaussians: Open-Vocabulary Scene Understanding with 3D Gaussian Splatting. *IEEE Transactions on Circuits and Systems for Video Technology*, 36 (7):9265–9277, 2026.

S. He, G. Jie, C. Wang, Y. Zhou, S. Hu, G. Li, and H. Ding. ReferSplat: Referring Segmentation in 3D Gaussian Splatting. In *International Conference on Machine Learning (ICML)*, 2025.

J. Hu, Z. Chen, Z. Li, Y. Xu, and J. Zhang. SparseLGS: Sparse View Language Embedded Gaussian Splatting. arXiv preprint, 2024.

B. Kerbl, G. Kopanas, T. Leimkühler, and G. Drettakis. 3D Gaussian Splatting for Real-Time Radiance Field Rendering. *ACM Transactions on Graphics (SIGGRAPH)*, 2023.

J. Kerr, C. M. Kim, K. Goldberg, A. Kanazawa, and M. Tancik. LERF: Language Embedded Radiance Fields. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 19729–19739, 2023.

A. Kirillov, E. Mintun, N. Ravi, H. Mao, C. Rolland, L. Gustafson, T. Xiao, S. Whitehead, A. C. Berg, W. Y. Lo, P. Dollár, and R. Girshick. Segment Anything. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, 2023.

H. Li, Y. Wu, J. Meng, Q. Gao, Z. Zhang, R. Wang, and J. 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 (CVPR)*, 2025.

S. Liang, S. Wang, K. Li, M. Niemeyer, S. Gasperini, H. P. A. Lensch, N. Navab, and F. Tombari. SuperGSeg: Open-Vocabulary 3D Segmentation with Structured Super-Gaussians. In *2026 International Conference on 3D Vision (3DV)*, pp. 1650–1659, 2026.

F. Liu, Y. Liu, K. Xu, G. Hancke, and R. Lau. GenSplat: Bridging the Generalization Gap in 3DGS Language Comprehension. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2026a.

Y. Liu, W. Li, C. Zhu-Tian, and H. Pfister. LangFlash: Feed-forward 3D Language Gaussian Splatting from Sparse Unposed Images. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) Findings*, 2026b.

Z. Liu, Y. Wang, S. Zheng, T. Pan, L. Liang, Y. Fu, and X. Xue. ReasonGrounder: LVLM-Guided Hierarchical Feature Splatting for Open-Vocabulary 3D Visual Grounding and Reasoning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2025.

T. Lu, M. Yu, L. Xu, Y. Xiangli, L. Wang, D. Lin, and B. Dai. Scaffold-GS: Structured 3D Gaussians for View-Adaptive Rendering. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2024.

Y. Lu, Y. Zhou, Y. Qiao, C. Song, T. Liang, J. Ma, H. Wang, and Y. Yin. Segment Then Splat: Unified 3D Open-Vocabulary Segmentation via Gaussian Splatting. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2025.

Q. Peng, B. Planche, Z. Gao, M. Zheng, A. Choudhuri, T. Chen, C. Chen, and Z. Wu. 3D Vision-Language Gaussian Splatting. In *International Conference on Learning Representations (ICLR)*, 2025.

Y. Peng, H. Wang, Y. Liu, C. Wen, Z. Dong, and B. Yang. GAGS: Granularity-Aware Feature Distillation for Language Gaussian Splatting. In *Proceedings of the AAAI Conference on Artificial Intelligence (AAAI)*, 2026.

J. Piekenbrinck, C. Schmidt, A. Hermans, et al. OpenSplat3D: Open-Vocabulary 3D Instance Segmentation using Gaussian Splatting. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPR Workshops)*, 2025.

M. Qin, W. Li, J. Zhou, H. Wang, and H. Pfister. LangSplat: 3D Language Gaussian Splatting. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 20051–20060, 2024.

J.-C. Shi, M. Wang, H.-B. Duan, and S.-H. Guan. Language Embedded 3D Gaussians for Open-Vocabulary Scene Understanding. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2024.

L. Tian, X. Li, L. Ma, H. Yin, Z. Zheng, H. Huang, T. Li, H. Lu, and X. Jia. CCL-LGS: Contrastive Codebook Learning for 3D Language Gaussian Splatting. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, 2025.

Y. J. Tseng, C. H. Kao, J. Z. Chen, A. Gnutti, S. Y. Lo, Y. Y. Lin, and W. H. Peng. CSGaussian: Progressive Rate-Distortion Compression and Segmentation for 3D Gaussian Splatting. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision (WACV)*, 2026.

Y. Wang, Z. Li, L. Guo, W. Yang, A. C. Kot, and B. Wen. ContextGS: Compact 3D Gaussian Splatting with Anchor Level Context Model. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2024.

Y. Wu, J. Meng, H. Li, et al. OpenGaussian: Towards Point-Level 3D Gaussian-based Open Vocabulary Understanding. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2024.

Y. Xie, A. Arafa, A. Javanmardi, et al. ReLaGS: Relational Language Gaussian Splatting. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 23826–23836, 2026.

B. Xiong, R. Liu, K. Xu, M. Chen, and A. Feng. Splat Feature Solver. In *International Conference on Learning Representations (ICLR)*, 2026.

M. Ye, M. Danelljan, F. Yu, and L. Ke. Gaussian Grouping: Segment and Edit Anything in 3D Scenes. In *Proceedings of the European Conference on Computer Vision (ECCV)*, 2024.

H. Zhai, Q. Zhang, X. Pan, X. Zhang, Y. Dong, H. Zhang, D. Xu, and G. Zhang. OnlinePG: Online Open-Vocabulary Panoptic Mapping with 3D Gaussian Splatting. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2026.

C. Zhang and G. H. Lee. EconSG: Efficient and Multi-View Consistent Open-Vocabulary 3D Semantic Gaussians. In *International Conference on Learning Representations (ICLR)*, 2025.

Z. Zhang, T. Wang, Z. Xu, Q. Ji, X. Wang, and Z. Zhan. Semantic-Consistent 3D Reconstruction via Gaussian Splatting and SAM-Guided Annotation. In *The International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences*, 2025.

S. Zhou, H. Chang, S. Jiang, Z. Fan, Z. Zhu, D. Xu, P. Chari, S. You, Z. Wang, and A. 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 (CVPR)*, 2024.

A APPENDIX

A.1 FULL RESULTS ON THE SCANNET DATASET

The main results are reported on LERF-OVS, which evaluates on rendered 2D images and focuses on pixel-level segmentation under novel views. To cover a genuinely three-dimensional evaluation dimension, we further report results on the large-scale indoor dataset ScanNet (Dai et al., 2017). ScanNet evaluates per-point in 3D point cloud space, where the output representation directly corresponds to the 3D GT point cloud, forming a complementary 2D and 3D pair with LERF-OVS.

The standard evaluation protocol for ScanNet was introduced by OpenGaussian (Wu et al., 2024) and has been widely adopted by subsequent work. This protocol requires initialization with the officially provided point cloud, freezing point coordinates during training and disabling densification, so that the number and coordinates of output points are strictly consistent with the GT point cloud, enabling per-point correspondence for computing mIoU and mAcc. This constraint is structurally incompatible with the anchor mechanism of Scaffold-GS. After anchor coordinates are frozen, each anchor still needs to generate at least $k \geq 2$ inner Gaussians through the decoding network, resulting in an output Gaussian count several times that of the GT points, making per-point correspondence with the GT impossible. When $k = 1$, meaning one anchor corresponds to only one Gaussian, the anchor and its decoding mechanism degenerate and no longer carry structural meaning. Therefore, on ScanNet we directly treat the officially provided points as native Gaussians, with one point corresponding to one Gaussian, and then apply the endogenous semantic mapping described in Section 3. Only the first-layer input dimension of the semantic network is adapted from the 35 dimensions of anchors to the 59 dimensions of native Gaussian attributes; the remaining network structure, 16-dimensional semantic latent space, rendering alignment, and loss settings are identical to those in the main text.

This adaptation enables our method to satisfy the standard ScanNet protocol while demonstrating that endogenous semantic mapping does not depend on a specific anchor structure and holds equally well on native 3DGS. It should be noted that the ScanNet protocol fixes the output point count and cannot be used for representation sparsity sweeps; our sparsity analysis is conducted on LERF-OVS, where densification is allowed.

We evaluate under the standard ScanNet open-vocabulary semantic segmentation protocols of 19, 15, and 10 classes, reporting both mIoU and mAcc, with results shown in Table A1.

Table A1: Open-vocabulary semantic segmentation results on the ScanNet dataset (%)

Method	19-cls mIoU	19-cls mAcc	15-cls mIoU	15-cls mAcc	10-cls mIoU	10-cls mAcc
LangSplat (Qin et al., 2024) (CVPR 2024)	3.78	9.11	5.35	13.20	8.40	22.06
OpenGaussian (Wu et al., 2024) (NeurIPS 2024)	24.73	41.54	30.13	48.25	38.29	55.19
LaGa (Cen et al., 2025b) (ICML 2025)	32.50	49.10	35.50	53.50	42.60	63.20
InstanceGaussian (Li et al., 2025) (CVPR 2025)	**40.66**	<u>54.01</u>	**42.51**	59.15	<u>47.94</u>	64.01
THGS (Dai et al., 2025) (ACM MM 2025)	34.39	50.74	39.61	57.07	46.38	64.74
ReLaGS (Xie et al., 2026) (CVPR 2026)	32.35	44.52	40.04	<u>60.59</u>	47.17	<u>66.08</u>
Ours	<u>40.33</u>	**61.78**	<u>41.43</u>	**62.56**	**48.73**	**69.72**

Note: Baseline results are taken from the publicly reported values in their respective original papers. Our results are measured under the standard 10-scene, 19/15/10-class protocols. Both mIoU and mAcc use the point cloud segmentation metric, where mAcc is the mean per-class point accuracy. Among the listed methods, bold indicates the best and underline indicates the second best.

A.2 IMPLEMENTATION DETAILS

The 2D pseudo-labels are generated by frozen foundation models. A multimodal large model first performs initial localization on the training images based on text queries, and the bounding boxes are then fed into SAM 3.1 to obtain masks with precise boundaries. The masked regions are encoded by OpenCLIP ViT-L/14 (D=768) to extract embeddings, which serve as the alignment targets for endogenous semantic mapping. This pipeline is consistent with the 2D supervision approach used by all comparison methods.

A.3 ABLATION STUDIES

We conduct three sets of controlled-variable ablations to examine the effects of semantic mapping network width, the two-stage training strategy, and the input composition on segmentation performance.

All ablations fix $k = 10$, and except for the variable under investigation, the training iterations, loss weights, optimizer, and pseudo-labels remain unchanged.

A.3.1 Semantic Mapping Network Width Ablation

Beyond the number of representation units, whether the width of the semantic mapping network itself also contains capacity redundancy is one aspect of the representation efficiency question. This ablation fixes $k = 10$ and keeps all settings including training iterations, loss weights, optimizer, and pseudo-labels unchanged, only changing the width W of the largest intermediate layer of the semantic MLP from 1024 to 768 and 512. The rest of the network structure ($35 \to 128$ and $128 \to 16$) remains unchanged. Table A2 reports the corresponding MLP parameter counts, relative ratios, and segmentation accuracy.

Table A2: Semantic MLP intermediate layer width ablation ($k = 10$, LERF-OVS average)

Intermediate width W	MLP parameters	Relative ratio	mIoU (%)	mAcc (%)
1024	269,968 (~270K)	100%	69.81	85.59
768	204,176 (~204K)	75.63%	67.92	86.36
512	138,384 (~138K)	51.26%	69.67	90.20

Note: Parameter counts are computed precisely from the network structure and include biases of all fully connected layers. The four-layer widths are $35 \to 128 \to W \to 128 \to 16$. This count only includes the semantic mapping MLP and does not include the subsequent two-layer 1×1 convolution dimension alignment module.

A.3.2 Two-Stage Training Ablation

To quantify the actual contribution of the second-stage SFNS fine-tuning, this ablation compares the first-stage result using only the base embedding loss with the full two-stage result incorporating SFNS. Both share the same first-stage weights and all other settings. Results are shown in Table A3.

Table A3: Two-stage training ablation ($k = 10$, LERF-OVS average)

Training setup	mIoU (%)	mAcc (%)
First stage only ($\mathcal{L}_{\text{embed}}$ only)	56.35	81.88
Full two stages ($\mathcal{L}_{\text{embed}}$ + SFNS)	69.81	85.59

Note: The first stage using only the embedding loss achieves only soft alignment with the CLIP space. Direct thresholding of cosine similarity yields segmentation with blurred boundaries and uncalibrated foreground/background, resulting in low mIoU. The second stage introduces discriminative fine-tuning with BCE and Dice, converting soft alignment into features that can be directly segmented.

A.3.3 Input Composition Ablation

The input to the semantic mapping is formed by concatenating the 32-dimensional anchor structural features with the 3-dimensional spatial coordinates. To distinguish the respective contributions, this ablation retains only the 32-dimensional structural features or only the 3-dimensional spatial coordinates, comparing each with the full 35-dimensional input. The first-layer input dimension of the MLP is adjusted accordingly, while the remaining network structure and training settings are unchanged. Results are shown in Table A4.

Table A4: Input composition ablation ($k = 10$, LERF-OVS average)

Input composition	Input dim	mIoU (%)	mAcc (%)
Full (structural features + coordinates)	35	69.81	85.59
Structural features only	32	66.55	87.50
Spatial coordinates only	3	60.93	80.83

Note: When changing the input, the first-layer dimension of the MLP is adjusted from 35 to 32 or 3 accordingly, while the remaining three layers and training settings are unchanged. Removing structural features and keeping only coordinates reduces mIoU to 60.93; removing coordinates and keeping only structural features reduces it to 66.55. Both are below the full input's 69.81, indicating that structural features and coordinates both make substantial contributions, with structural features contributing more and the spatial priors provided by coordinates also accounting for a considerable portion of categories in indoor scenes.

A.4 Full Sparsity Sensitivity Results (Corresponding to Figure 2 in the Main Text)

To provide the specific values corresponding to the curves in Figure 2, we report results by anchor generation parameter k, covering $k = 2$, 5, 10, 15, and 20. For each setting, we report semantic accuracy (mIoU and mAcc), anchor count, rendered Gaussian count, and rendering quality PSNR for each of the four scenes. Each table corresponds to one k and contains the four LERF-OVS scenes. The last row of each table reports the arithmetic mean across the four scenes.

Table A5: Results for $k = 2$

Scene	mIoU (%)	mAcc (%)	Anchor count	Rendered Gaussian count	PSNR (dB)
Figurines	66.78	85.71	155,953	311,906	23.40
Teatime	78.94	93.22	347,676	695,352	24.43
Ramen	65.67	83.10	234,526	469,052	25.53
Kitchen	65.17	81.82	456,347	912,694	24.97
Mean	69.14	85.96	298,626	597,251	24.58

Table A6: Results for $k = 5$

Scene	mIoU (%)	mAcc (%)	Anchor count	Rendered Gaussian count	PSNR (dB)
Figurines	67.59	85.71	179,710	898,550	23.31
Teatime	73.57	98.31	379,566	1,897,830	24.54
Ramen	65.77	87.32	235,899	1,179,495	25.65
Kitchen	63.38	81.82	438,135	2,190,675	25.28
Mean	67.58	88.29	308,328	1,541,638	24.70

Table A7: Results for $k = 10$

Scene	mIoU (%)	mAcc (%)	Anchor count	Rendered Gaussian count	PSNR (dB)
Figurines	70.50	83.93	167,587	1,675,870	23.47
Teatime	77.07	94.92	354,935	3,549,350	24.64
Ramen	68.00	81.69	189,597	1,895,970	25.57
Kitchen	63.66	81.82	371,500	3,715,000	24.76
Mean	69.81	85.59	270,905	2,709,048	24.61

Table A8: Results for $k = 15$

Scene	mIoU (%)	mAcc (%)	Anchor count	Rendered Gaussian count	PSNR (dB)
Figurines	65.69	85.71	166,184	2,492,760	23.33
Teatime	80.13	96.61	356,008	5,340,120	24.51
Ramen	68.59	85.92	198,018	2,970,270	25.65
Kitchen	65.28	77.27	331,734	4,976,010	25.24
Mean	69.92	86.38	262,986	3,944,790	24.68

Table A9: Results for $k = 20$

Scene	mIoU (%)	mAcc (%)	Anchor count	Rendered Gaussian count	PSNR (dB)
Figurines	67.31	83.93	168,697	3,373,940	23.24
Teatime	75.17	98.31	332,482	6,649,640	24.79
Ramen	68.51	85.92	184,511	3,690,220	25.35
Kitchen	64.89	86.36	284,260	5,685,200	25.02
Mean	68.97	88.63	242,488	4,849,750	24.60

A.5 Runtime Resource Cost

Table 2 in the main text compares peak GPU memory across methods on the Figurines scene. This section further provides the complete runtime resources for the four scenes under the main $k = 10$ configuration, including allocated memory, nvidia-smi peak memory, and total two-stage training time, as shown in Table A10.

Table A10: Runtime resources for each scene at $k = 10$

Scene	Allocated memory (GiB)	nvidia-smi memory (GiB)	Training time (min)
Figurines	1.52	7.09	22.57
Teatime	2.47	8.89	24.64
Ramen	2.00	9.46	25.33
Kitchen	3.55	13.15	24.99

Note: Allocated memory uses the torch.cuda.max_memory_allocated() metric, which only counts memory actually allocated to tensors and excludes caching allocator reserves and the CUDA context. nvidia-smi memory is the total process occupancy, including PyTorch caching allocator reserves, CUDA context, and geometry data, consistent with the metric used in Table 2 of the main text and by comparison baselines. Values in the two columns are not directly interchangeable. Training time is the total duration of two-stage semantic training on a single RTX 4090.

A.6 LIMITATIONS

This work is positioned as an empirical study at the representation level rather than an engineering system pursuing maximum accuracy. The main limitation is that our conclusions rely on a single representation platform. Both the redundancy analysis and quantitative results are built on Scaffold-GS structured anchors, without repeating experiments on native 3DGS or other sparse or hierarchical Gaussian variants. Scaffold-GS was chosen to fully preserve geometric structure when reducing representation counts, so that changes in accuracy can be cleanly attributed to semantic redundancy. However, this also means that the specific finding that roughly 10% of units can support semantics has not been verified on native dense 3DGS, other sparsification mechanisms, or in outdoor, street-view, and large-scale scenes. This work provides evidence on representation scale and design direction rather than a universal constant across platforms.