⚡LightSplat: Fast and Memory-Efficient Open-Vocabulary 3D Scene Understanding in Five Seconds

Jaehun Bang[1] Jinhyeok Kim[2*] Minji Kim[1] Seungheon Jeong[1] Kyungdon Joo[1†]
[1] AIGS, UNIST [2] GSAI, POSTECH
{devappendcbangj, mzkim, sypsss, kyungdon}@unist.ac.kr jh4011@postech.ac.kr

Abstract

Open-vocabulary 3D scene understanding enables users to segment novel objects in complex 3D environments through natural language. However, existing approaches remain slow, memory-intensive, and overly complex due to iterative optimization and dense per-Gaussian feature assignments. To address this, we propose `LightSplat`*, a fast and memory-efficient training-free framework that injects compact 2-byte semantic indices into 3D representations from multi-view images. By assigning semantic indices only to salient regions and managing them with a lightweight index-feature mapping,* `LightSplat` *eliminates costly feature optimization and storage overhead. We further ensure semantic consistency and efficient inference via single-step clustering that links geometrically and semantically related masks in 3D. We evaluate our method on LERF-OVS, ScanNet, and DL3DV-OVS across complex indoor-outdoor scenes. As a result,* `LightSplat` *achieves state-of-the-art performance with up to 50-400× speedup and 64× lower memory, enabling scalable language-driven 3D understanding. For more details, visit our project page* https://vision3d-lab.github.io/lightsplat/.

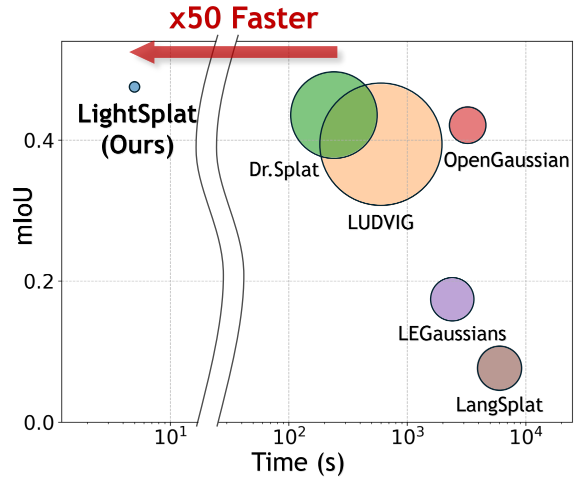

Figure 1. **Comprehensive comparison of speed, performance, and memory overhead.** We evaluate recent open-vocabulary 3D scene understanding models in terms of distillation time (x-axis), segmentation performance (y-axis), and memory overhead (circle size). `LightSplat` achieves 50× faster feature distillation, higher accuracy, and 64× lower memory usage. LUDVIG's circle is shown at half size because it is too large to display at full scale.

1. Introduction

With growing demand for natural user interactions within 3D environments, open-vocabulary 3D scene understanding has emerged as an important task [1, 10, 12, 17, 20, 22, 27]. It aims to identify and segment 3D scenes based on user queries, without predefined categories. This flexibility allows handling complex environments with diverse and unpredictable objects, enabling real-world applications in robotic manipulation [8], 3D scene editing [7], and AR/VR.

A main challenge in this task is bridging the gap between language and 3D representations. To address this, recent approaches distill 2D semantics into the 3D scene using object cues from SAM [13] and semantic embeddings from CLIP [21]. As an early attempt, LERF [12] embeds CLIP features into NeRF [18], but suffers from limited interactivity and scalability due to its implicit representation and high computational cost. To enable efficient language-to-3D alignment, 3D Gaussian Splatting (3DGS) [11] provides an explicit representation with real-time rendering. Leveraging this representation, recent approaches [1, 10, 17, 20, 22, 27] distill language semantics into the 3D scene. These methods typically rely on rendering-guided iterative feature optimization, contrastive learning for semantic separation, and feature quantization to compress language features.

Despite recent advances, existing methods still suffer from three major limitations: high computational cost, memory overhead, and semantic degradation, all of which hinder scalability in real-world scenarios. First, feature dis-

[*] Work done while at UNIST.
[†] Corresponding author.

tillation is bottlenecked by iterative optimization that repeatedly aligns rendered views with CLIP embeddings. Although a direct mapping between 2D masks and 3D structure is feasible, this pipeline remains unnecessarily inefficient. Second, assigning high-dimensional features to individual Gaussians leads to redundant storage and excessive per-Gaussian comparisons during inference. Third, semantic degradation arises when Gaussians are projected into 2D, which blurs their features and leads to indirect supervision misaligned with 3D geometry.

We propose `LightSplat`, a fast and memory-efficient training-free framework for 3D scene understanding via lightweight semantic injection. Our key insight is to bypass costly optimization by directly linking 2D semantics to 3D structure through discrete indexing. In our method, we inject semantics only into Gaussians that have a high rendering contribution to the corresponding 2D masks. Rather than storing full language features per Gaussian, we assign compact 2-byte mask indices, which are mapped to CLIP features via a lightweight mask-feature mapping. This decouples feature storage from the 3D representation, eliminating per-Gaussian feature handling and significantly reducing memory and computational overhead. To further improve inference speed and ensure semantic consistency in 3D, we cluster Gaussians based on geometric and semantic relationships, forming visually and conceptually meaningful objects. By consolidating 2D mask features into 3D cluster representations using precomputed relationships among the 2D masks, we achieve fast and consistent object-level inference. We evaluate `LightSplat` on LERF-OVS, ScanNet, and DL3DV-OVS, an open-vocabulary extension of DL3DV [16] for complex indoor-outdoor scenes. With the streamlined design, `LightSplat` distills features in only 5 seconds, up to 50-400× faster than the previous state-of-the-art method [10], while outperforming it in both efficiency and performance as illustrated in Fig. 1. Moreover, our method reduces memory usage by 64× with object-level indexed features instead of Gaussian-level language features. This design provides a lightweight and scalable foundation for real-world scenarios [5, 12, 16].

In summary, our main contributions are as follows:

- We propose `LightSplat`, a simple, training-free framework for open-vocabulary 3D scene understanding eliminating exhaustive iterative optimization.
- Our approach assigns each Gaussian a small index tied to object semantics via an index-feature mapping, enabling fast inference without per-Gaussian feature handling.
- We design a geometry and semantic-aware clustering method that constructs object-level representations in 3D, enabling efficient and interpretable inference.
- Our method distills semantics in 5 seconds, achieving 50-400× speedup and 64× lower memory than prior SOTA, while preserving high-quality 3D semantics.

2. Related Work

3D Scene Representations. 3D scene representations reconstruct 3D environments from multi-view images, facilitating novel view synthesis and 3D scene understanding. NeRF provides high-quality rendering via MLP-based volumetric fields, but remains slow and inefficient for real-world applications despite efforts to accelerate it [6, 19, 25, 28, 29]. To overcome these computational bottlenecks, 3DGS has emerged as a promising alternative, offering real-time rendering and explicit scene representation. It models a scene using a collection of 3D Gaussian primitives, which can be efficiently rendered through differentiable rasterization. As an explicit representation, 3DGS offers higher interpretability and direct access to its primitives, enabling easier integration into various applications [3, 9, 26, 30].

In this work, we leverage the real-time and explicit nature of 3DGS to introduce an efficient and straightforward method for injecting semantics into 3D scenes.

Open-Vocabulary 3D Scene Understanding. Given a natural language query, the goal of open-vocabulary 3D scene understanding is to segment corresponding objects in a 3D scene. Without relying on predefined categories, this approach enables more flexible and scalable scene understanding in real-world environments, where object categories are diverse. Many studies have explored integrating language features into 3D scene representation, focusing on distilling semantic embeddings from 2D foundation models like CLIP [21], DINO [2], and SAM [13] to the radiance field.

Early works [12, 14, 23, 24] introduce additional networks into NeRF to learn semantic features. DFF [14] augments the network for feature prediction, and LERF [12] uses multi-scale CLIP features as supervision. While these methods offer a unified 3D-language representation, they are limited by NeRF's slow and computationally intensive volumetric rendering.

To address this, recent studies [1, 15, 20, 22, 31] adopt 3DGS and optimize language features through rendering-based iterative training. This optimization gradually enriches the 3D Gaussians with language semantics, leading to interpretable 3D representations. For example, LangSplat [20] and LEGaussians [22] distill CLIP features and compress them using an autoencoder and codebooks. While these methods provide a useful representation, they suffer from slow feature distillation caused by iterative feature optimization. Moreover, because the segmentation happens on 2D rendered feature maps, the model cannot directly localize the corresponding objects in 3D space. OpenGaussian [27] performs segmentation in 3D space, but still relies on an iterative process to train features. Furthermore, its codebook-based feature quantization struggles to

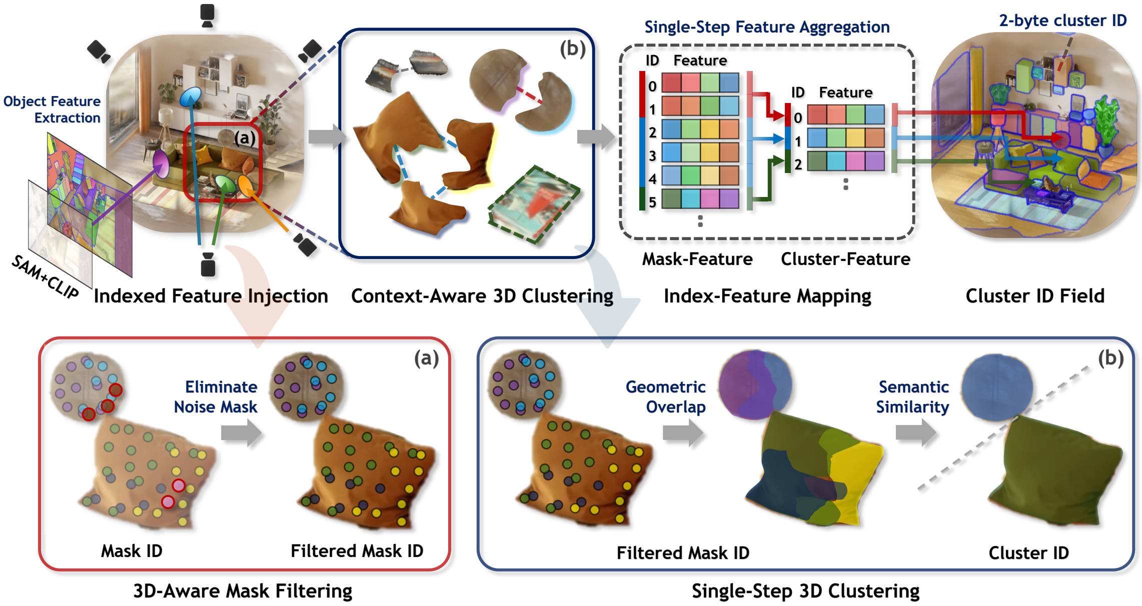

Figure 2. **Overall framework of LightSplat.** From multi-view images, we obtain SAM masks and corresponding CLIP features. We align them to the 3D scene via indexed feature injection, assigning each Gaussian a compact 2-byte mask index based on its projection influence. (a) To ensure semantic consistency, we perform 3D-aware mask filtering, and (b) construct an inter-mask graph via index-feature mapping based on geometric and semantic relations, which guides context-aware 3D clustering in a single step. This enables cluster-level feature management with a compact cluster ID field for efficient, interpretable, and training-free open-vocabulary 3D scene understanding.

fully cover the scene when the scene complexity exceeds the codebook's capacity.

Recent works such as Dr.Splat [10] and LUDVIG [17] associate CLIP features directly with 3D Gaussians by projecting 2D features onto them. However, both still rely on iterative feature aggregation and store high-dimensional features for every Gaussian, leading to unnecessary computational and memory cost. In addition, Dr.Splat requires large-scale Product Quantization training, while LUDVIG employs graph diffusion, further increasing computational overhead. These limitations highlight the need for a simpler semantic representation that stores information only where it is useful, reducing both memory use and computation. Motivated by these limitations, we propose a training-free and memory-efficient method that represents object-level semantics using compact indices, which are linked to features via a lightweight index-feature mapping.

3. Method

3.1. Overview

The key insight of LightSplat is that object semantics in 2D masks can be directly lifted to 3D via compact mask indices. This enables single-step semantic injection and inter-mask clustering without per-Gaussian features. As a result, it supports fast 2D-3D semantic alignment and scalable deployment without iterative training.

Fig. 2 illustrates the overall LightSplat framework.

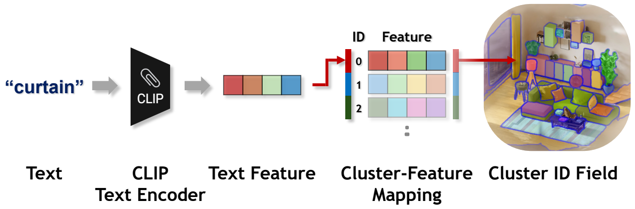

Figure 3. **Fast inference via cluster-feature mapping.** During inference, the text query is compared with a compact set of cluster features instead of all Gaussians or pixels, enabling fast retrieval.

The pipeline begins by extracting 2D object masks and their corresponding CLIP features from multi-view images using SAM and CLIP, following prior work [20]. Instead of storing high-dimensional features per Gaussian, we assign compact 2-byte mask indices based on their contribution to the 2D mask. To improve the reliability of this mapping, we filter out noisy masks that have limited impact on the 3D scene. This reduces view-dependent artifacts and strengthens multi-view semantic consistency. To manage semantics efficiently, we propose an index-feature mapping that associates each 2-byte index to its corresponding CLIP feature. This design replaces redundant per-Gaussian features with a compact object-level representation, allowing fast and memory-efficient inference. Finally, we perform context-aware 3D clustering to group Gaussians into object-level clusters. To achieve this, we construct a graph over the filtered 2D masks based on their geometric overlap in 3D and semantic similarity, and partition it in a single step. We then assign each 3D cluster a representative language feature, enabling compact and interpretable object-level inference, as illustrated in Fig. 3.

3.2. Index-Feature Mapping

To overcome the inefficiency of iterative per-Gaussian optimization, we replace mask-level semantic features with compact 2-byte indices. These indices are linked to CLIP features via a mask-feature mapping, allowing each Gaussian to store only a mask index instead of high-dimensional language features. Each mask index acts as a key to its language feature in the index-feature mapping tensor. During clustering, we consolidate the mask-feature mapping into a compact cluster-feature mapping by grouping related masks across multiple views in a single step. These mask-feature and cluster-feature mappings collectively form the index-feature mapping. This index-feature mapping is not merely a storage reduction technique. It serves as the core design that accelerates the pipeline, preserves semantic consistency across views, and enables scalable 3D understanding.

3.3. Indexed Feature Injection

This section describes a single-step transfer of 2D semantics into 3D while preserving semantic information and removing iterative optimization. To achieve this, we exploit the mask-feature mapping by decoupling feature storage from the 3D representation and injecting compact mask indices into the 3D scene.

Object Feature Extraction. We first extract object semantics from each view, which serves as a crucial foundation for composing 3D object semantics. Specifically, given multi-view images $\mathbf{I} = \{I_l\}_{l=1}^{L}$ where L is the total number of frames, each image $I_l \in \mathbb{R}^{3\times H\times W}$ is of size $H \times W$, with 3 color channels. For each view, we use SAM to extract object masks $\mathcal{M}$:

$$\mathcal{M} = \{m_k\}_{k=1}^{K}, \quad m_k \in \{0,1\}^{H\times W}, \tag{1}$$

where K is the total number of masks. For each mask, we extract the corresponding CLIP features:

$$\mathcal{F} = \{f_k\}_{k=1}^{K}, \quad f_k \in \mathbb{R}^{512}. \tag{2}$$

We then assign each 2D mask a unique index to manage its CLIP features and inject semantics efficiently into 3DGS.

Gaussian Contribution. To assign semantics only to Gaussians that significantly contribute to the rendered image, we compute their pixel-wise contributions using alpha-blending weights from the rendering equation [11]. The n-th Gaussian contribution at pixel (u, v) in the l-th view is defined as:

$$w_n^{(l)}(u,v) = \alpha_n \cdot T_n^{(l)}(u,v), \tag{3}$$

where α_n is the opacity of the n-th Gaussian. The transmittance $T_n^{(l)}(u,v)$ quantifies how much light reaches the n-th Gaussian at pixel (u, v) in the l-th view. It is computed by accumulating the occlusion effects from all preceding Gaussians along the same ray:

$$T_n^{(l)}(u,v) = \prod_{j=1}^{n-1} \left(1 - \alpha_j(u,v)\right), \tag{4}$$

where $\alpha_j(u,v)$ denotes the opacity of the j-th Gaussian at that pixel.

Indexed Feature Injection. To achieve efficient semantic injection, we assign 2-byte mask indices instead of full language features to Gaussians that contribute meaningfully in the image space:

$$\mathcal{G}_k = \left\{ g_n \,\middle|\, m_k^{(l)}(u,v) = 1,\ w_n^{(l)}(u,v) \geq \tau_{\text{contrib}} \right\}, \tag{5}$$

where $\mathcal{G}_k$ is the set of Gaussians g_n associated with mask m_k, with τ_{contrib} for contribution filtering. We assign semantics only to Gaussians that make a meaningful visual contribution, avoiding fixed-size selection (*e.g.*, top-k) that can include Gaussians with negligible visual impact. Our approach significantly reduces memory usage by 1024 times compared to storing CLIP features directly, reducing the size from 4×512 bytes to just 2 bytes per Gaussian. This enables lightweight mask-level feature management and scalable object-level reasoning in 3D clustering without accessing per-Gaussian features.

3D-Aware Mask Filtering. To enhance semantic reliability and efficiency, we filter out masks that contribute insufficiently to 3D feature construction:

$$\mathcal{M}_{\text{filtered}} = \{m_k \mid |\mathcal{G}_k| \geq \tau_{noise}\}, \tag{6}$$

where $|\mathcal{G}_k|$ denotes the number of Gaussians associated with the k-th SAM mask, measuring its geometric support in 3D. This step leverages Gaussian contributions from 2D-3D correspondences to suppress view-dependent artifacts while preserving masks with sufficient geometric support under τ_{noise}. Finally, we assign each Gaussian the most influential mask index across multiple views.

As a result, indexed feature injection compactly transfers 2D semantics into 3D based on each Gaussian contribution, forming the basis for scalable object-level reasoning.

3.4. Context-Aware 3D Clustering

To achieve more scalable and interpretable 3D understanding, we cluster Gaussians into object-level groups using semantic and spatial context. Leveraging the mask indices from the previous stage, our method first connects semantically related 2D masks across views. Their associated

Gaussians are then grouped into coherent 3D clusters without storing high-dimensional per-Gaussian features. This compact representation efficiently transfers 2D mask semantics into 3D coherent objects, enabling faster, memory-efficient, and consistent multi-view understanding.

Single-Step 3D Clustering. To enable object-level 3D clustering in a single step, we construct an undirected graph G, where each node V represents a filtered 2D mask $m_k \in \mathcal{M}_{\text{filtered}}$, and edges E represent connectivity between these masks:

$$G = (V, E), \quad V = \{\nu \mid m_k \in \mathcal{M}_{\text{filtered}}\}\,. \tag{7}$$

We initialize E with self-loop edges, defining $E = \{(k, k) \mid k \in V\}$, so that each node initially forms its own cluster. To determine mask connectivity in the graph, we measure their geometric overlap and semantic similarity between masks m_k and $m_{k'}$ in 3D space. First, we compute the geometric overlap using the Intersection over Union (IoU) of their corresponding Gaussian sets:

$$\text{IoU}\big(\mathcal{G}_k, \mathcal{G}_{k'}\big) = \frac{|\mathcal{G}_k \cap \mathcal{G}_{k'}|}{|\mathcal{G}_k \cup \mathcal{G}_{k'}|}. \tag{8}$$

Second, we compute the semantic similarity using the cosine similarity between their CLIP features:

$$\text{sim}(f_k, f_{k'}) = \frac{f_k \cdot f_{k'}}{\|f_k\|\,\|f_{k'}\|}. \tag{9}$$

Then, we connect masks k and k' with an edge if their geometric and semantic agreement suggests that they correspond to the same object:

$$(k, k') \in E \quad \text{if} \quad \begin{cases} \text{IoU}(\mathcal{G}_k, \mathcal{G}_{k'}) \geq \tau_{\text{IoU}} \\ \text{sim}(f_k, f_{k'}) \geq \tau_{\text{feat}} \end{cases}, \tag{10}$$

with τ_{IoU} and τ_{feat} as cutoffs for geometric overlap and semantic similarity, respectively. Each cluster corresponds to a connected component of G.

$$(k, k') \in E \quad \Longrightarrow \quad \mathcal{C}_k = \mathcal{C}_{k'}. \tag{11}$$

Single-Step Feature Aggregation. For efficient feature management, we compute the cluster-level semantic features in a single step by directly averaging the CLIP features of all associated masks. Since mask-feature mappings are already established and clustering operates at the mask-level, the cluster-feature mapping can be constructed immediately in this single step. For example, inference complexity can be reduced from comparing over 100,000 Gaussians to only about 100 clusters, lowering similarity computations to below 0.1% depending on scene complexity.

Overall, our clustering strategy provides view-consistent 3D semantics while enabling fast distillation, efficient inference, and compact memory usage through jointly leveraging geometric and semantic cues.

4. Experiments

4.1. Experimental Setup

Dataset and Baselines. To evaluate 3D scene understanding, we use three benchmarks: LERF-OVS [20], ScanNet [5], and DL3DV-OVS, covering both standard and more complex environments. The first is LERF-OVS, which consists of multi-view images with diverse text query-2D mask annotations for each scene. The second is ScanNet, a large-scale RGB-D dataset containing 1,500 indoor scenes, each with reconstructed point clouds and per-point semantic labels. For fair comparison on ScanNet, we follow the evaluation protocol of OpenGaussian and use the same 10 scenes. For robustness evaluation beyond limited indoor environments, we introduce the DL3DV-OVS dataset. It is an open-vocabulary extension of DL3DV-10K [16], which we annotate with 2D mask-text pairs to cover large and complex indoor-outdoor scenes. The dataset covers a wide range of object scales, distances, and scene complexities across four scenes (park, road, shop, and office), with categories containing varying numbers of instances.

Leveraging these benchmarks, we perform a comprehensive comparison against recent methods, including LangSplat [20], LEGaussians [22], OpenGaussian [27], Dr.Splat [10], and LUDVIG [17]. Since Dr.Splat does not provide inference code, we adopt the reported inference results from its paper and measure all other results ourselves.

3D Understanding Tasks. We evaluate performance on two standard tasks. The first is 3D Object Selection, which aims to segment 3D objects based on open-vocabulary user queries. For this evaluation, we use the LERF-OVS and DL3DV-OVS dataset, which offer 2D mask-text pairs. Following previous works [10, 27], we identify the 3D Gaussians corresponding to a given text query and render them into 2D. The rendered results are evaluated using mIoU and mAcc@0.25 against the ground-truth masks.

The second is 3D Semantic Segmentation. In this task, each 3D Gaussian is classified with the most relevant semantic label, selected from a pool of open-vocabulary text inputs. Using ScanNet per-point semantic annotations, we evaluate directly in 3D with mIoU and mAcc, and report performance under the 19, 15, and 10 class settings.

4.2. 3D Object Selection

Quantitative Results. Even without training, our method achieves SOTA performance on LERF-OVS, with a 50× speedup over recent models, as shown in Table 1. In terms of accuracy, we outperform the previous SOTA by an average of 4 mIoU and 4.45 mAcc@0.25. Notably, our method performs well on challenging scenes like ramen, which contain small and complex objects such as eggs and chopsticks.

As shown in Table 2, our approach also achieves strong performance on DL3DV-OVS, a dataset with large and

Table 1. **Quantitative comparison for 3D Object Selection on the LERF-OVS dataset.** Red and orange highlight the best and second-best results in each column. FD Time refers to feature distillation time. If FD Time exceeds 20 minutes, it is rounded at the ones place.

Methods	mIoU					mAcc @ 0.25					**FD Time**
	Waldo	Ramen	Figurines	Teatime	**Mean**	Waldo	Ramen	Figurines	Teatime	**Mean**	
LangSplat	9.61	3.49	9.64	7.91	7.66	9.09	5.63	14.29	8.47	9.37	100 min
LEGaussians	17.21	14.15	16.40	21.93	17.42	27.27	28.17	25.00	33.90	28.56	40 min
OpenGaussian	30.96	24.02	55.38	58.24	42.15	45.45	28.17	75.00	76.27	56.22	50 min
LUDVIG	28.08	29.33	47.43	52.28	39.28	51.33	50.36	67.02	85.72	63.61	10 min
Dr.Splat	39.07	24.70	53.36	57.20	43.58	63.64	35.21	80.36	76.27	63.87	4 min
Ours	34.95	45.07	50.63	59.65	**47.58**	59.09	57.75	76.79	79.66	**68.32**	**4.2 s**

Table 2. **Quantitative comparison for 3D Object Selection on the DL3DV-OVS dataset.** Red and orange highlight the best and second-best results in each column. FD Time refers to feature distillation time. If FD Time exceeds 20 minutes, it is rounded at the ones place.

Methods	mIoU					mAcc @ 0.25					**FD Time**
	Park	Shop	Road	Office	**Mean**	Park	Shop	Road	Office	**Mean**	
LangSplat	8.79	9.28	7.90	15.80	10.44	0.00	17.65	16.67	27.27	15.40	220 min
LEGaussians	21.95	9.70	3.51	8.17	10.83	33.33	11.76	0.00	0.00	11.27	40 min
OpenGaussian	29.65	19.51	41.59	6.78	24.38	41.67	35.29	66.67	0.00	35.91	60 min
LUDVIG	37.03	33.56	27.68	18.58	29.21	70.00	56.75	53.57	47.22	56.89	12 min
Ours	69.78	35.59	31.43	43.13	**44.98**	91.67	47.06	50.00	54.55	**60.82**	**4.8 s**

Table 3. **Quantitative comparison for 3D Semantic Segmentation on the ScanNet dataset.** Red and orange highlight the best and second-best results in each column. FD Time, Runtime, and Memory indicate the feature distillation time, average inference time per text query, and feature size per Gaussian, respectively. If FD Time exceeds 20 minutes, it is rounded at the ones place.

Methods	19 classes		15 classes		10 classes		**FD Time**	**Runtime (second)**	**Memory (byte)**
	mIoU	mAcc	mIoU	mAcc	mIoU	mAcc			
LangSplat	2.61	10.11	4.08	13.22	6.30	20.48	40 min	2.1	36
LEGaussians	1.62	7.26	5.72	14.23	9.84	19.13	50 min	1.7	32
OpenGaussian	29.43	43.62	32.61	48.26	41.29	56.42	30 min	0.003	24
LUDVIG	28.47	44.17	31.47	48.54	40.47	58.15	4 min	0.006	2048
Dr.Splat	28.00	44.60	38.20	60.40	47.20	68.90	3 min	-	128
Ours	37.11	58.66	39.78	60.91	47.78	68.21	**4.1 s**	**0.002**	**2**

complex scenes containing multiple objects. Our method shows robust performance on the road scene with multiple distant cars and the office scene with repeated objects such as chairs and monitors. Compared to LUDVIG, it achieves 15.77 higher mIoU, 3.93 higher mAcc@0.25, and 150× faster feature distillation by eliminating iterative feature optimization. This capacity stems from merging object masks and grouping their associated Gaussians into object-level semantics. This prevents semantic blurring and preserves clean boundaries even in cluttered or large scenes. Such results highlight the flexibility and robustness of our method across diverse object scales and scene complexities.

Qualitative Results. Fig. 4 and Fig. 5 present the qualitative comparison on the LERF-OVS and DL3DV-OVS dataset. Rendering-based methods such as LEGaussians and LangSplat struggle to produce clean segmentations in open-vocabulary scenarios because semantics become blurred when Gaussians are mixed during 2D rendering. OpenGaussian associates language features in 3D space, but the lack of semantic guidance during clustering leads to inaccurate Gaussian grouping and inconsistent object boundaries. In contrast, our method produces clean segmentations even for small or complex objects, such as eggs and tea in a glass. This is achieved by jointly leveraging 2D mask semantics and their corresponding 3D geometry to guide clustering. Our method remains effective in DL3DV-OVS, maintaining clean boundaries across diverse challenges. It handles multiple target objects such as chairs and monitors, distant outdoor objects like cars, and large indoor areas with sizable objects like white sofas.

4.3. 3D Semantic Segmentation

Quantitative Results. Table 3 presents the quantitative results on the ScanNet dataset. Our method achieves the best performance across all settings. In the most challenging 19-class configuration, it surpasses the previous SOTA

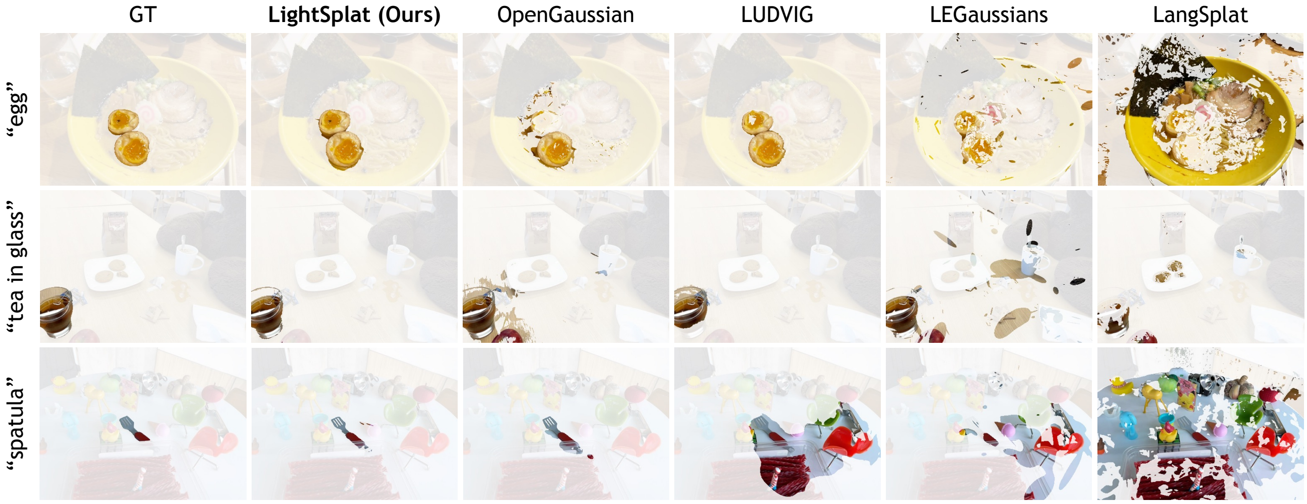

Figure 4. **Qualitative comparison for 3D Object Selection on the LERF-OVS dataset.** We visualize model performance across different scenes and text queries in LERF-OVS. With context-aware 3D clustering, our method achieves detailed object boundaries while offering significantly faster performance than other methods.

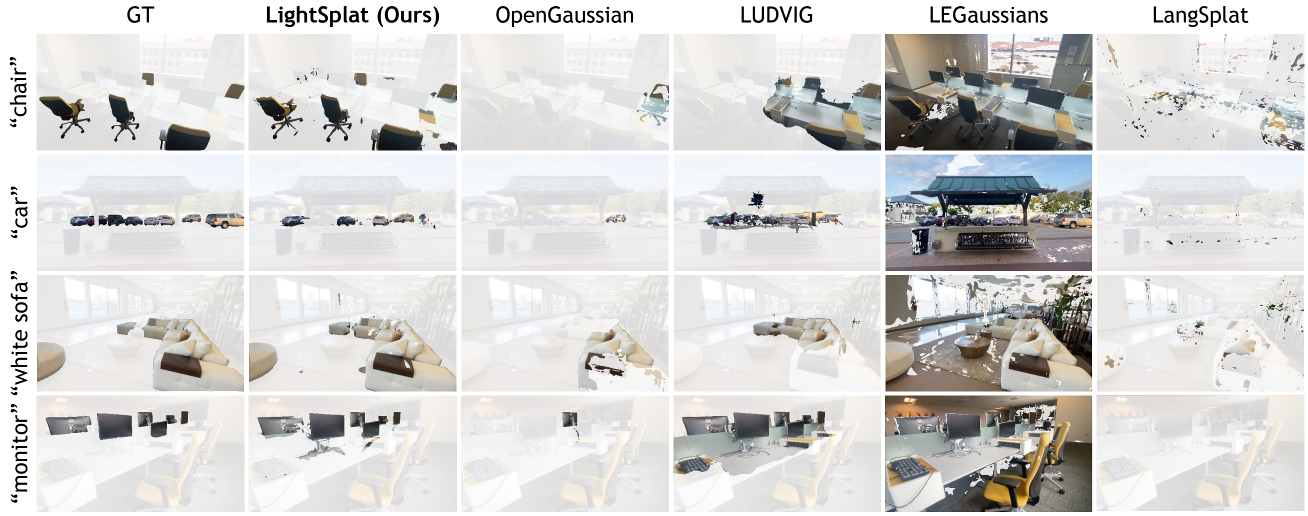

Figure 5. **Qualitative comparison for 3D Object Selection on the DL3DV-OVS dataset.** We visualize model behavior on large and complex scenes in DL3DV-OVS, covering both indoor and outdoor environments. Our method provides reliable selections and clear object boundaries, even in scenes with many similar objects.

by 7.68 mIoU and 14.06 mAcc, highlighting its strong 3D segmentation capability. In addition, our approach reduces feature distillation to only 4.1 seconds and improves memory efficiency by up to 64×, while supporting fast inference at 500 FPS for text-query retrieval using precomputed cluster features. These advantages stem from avoiding iterative per-Gaussian feature aggregation and high-dimensional feature handling, which are major bottlenecks in LUDVIG and Dr.Splat. Instead, our index-based design groups Gaussians into object-level units using compact semantic indices, enabling rapid feature distillation and real-time inference. These results further show that our method remains robust across text queries from object-centric descriptions to indoor spatial semantics, delivering fast and scalable performance for practical 3D applications.

Qualitative Results. Fig. 6 presents the qualitative results on the ScanNet dataset. Rendering-based methods like LEGaussians and LangSplat identify only a few object classes, as 2D semantic distillation struggles to separate objects in 3D space. OpenGaussian and LUDVIG better distinguish object regions but still suffer from blurry boundaries and mislabeling. Since these methods use semantics at the level of individual Gaussians, they fail to form meaningful object structure, making object-level reasoning diffi-

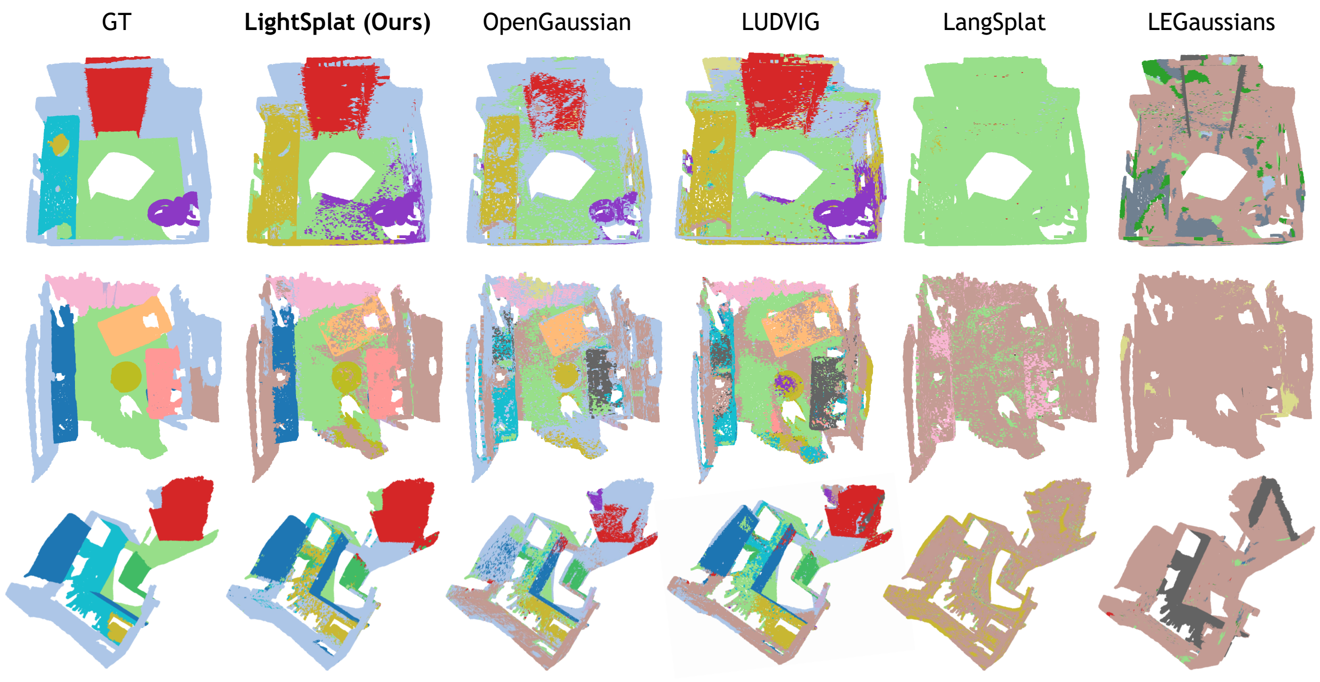

Figure 6. **Qualitative comparison for 3D Semantic Segmentation on the ScanNet dataset.** For intuitive comparison, ground-truth semantics are visualized in distinct colors, and predictions from each model are shown side by side. Compared to other methods, our approach more effectively captures both object (e.g., door) and large-area semantics (e.g., wall), demonstrating robustness across diverse real-world scenes and adaptability to a wide range of queries.

Table 4. **Ablation Study on LERF-OVS dataset.** The results demonstrate that each component not only boosts performance, but also directly contributes to the fast FD time of our method. FD Time refers to feature distillation time.

Ablation	mIoU	Acc@0.25	FD Time
Ours Full	47.58	68.32	4.55 s
w/o 3D Mask Filtering	29.31	25.96	4.54 s
w/o Semantic-Aware	19.56	18.97	4.42 s
w/o Geometry-Aware	2.01	2.27	4.44 s

cult. In contrast, our method captures semantics more consistently, handling both object-level categories (e.g. doors) and large spatial regions (e.g. walls and floors).

4.4. Ablation Study

We conduct an ablation study by removing each component individually. As shown in Table 4, each component significantly contributes to the overall performance. Without 3D-aware mask filtering, noisy masks degrade clustering quality, disrupting 2D-3D alignment and reducing mIoU from 47.58 to 29.31. Removing semantic-aware clustering decreases performance by over 50%, as the model cannot identify semantically corresponding masks across views for merging. Without geometry-aware clustering, the model ignores mask overlap in 3D, which prevents it from leveraging 2D-3D correspondences or maintain spatial consistency. As a result, performance drops sharply to 2.01 mIoU.

Notably, the feature distillation time is almost the same across all variants, differing by only about 0.1 seconds. This reflects the efficiency and necessity of our design. Most of the computation lies in evaluating Gaussian contributions to the 2D masks, which is inherently required, while the remaining steps add only negligible overhead.

5. Conclusion

We have proposed LightSplat, a fast and memory-efficient training-free framework for open-vocabulary 3D scene understanding. By directly injecting compact 2-byte mask indices into 3D Gaussians based on their influence, and managing semantics through an index-feature mapping, our method eliminates the need for iterative optimization and high-dimensional per-Gaussian storage. Furthermore, our context-aware clustering uses geometric and semantic cues to form cluster-level semantics from per-mask features, enabling fast and interpretable inference. Extensive experiments demonstrate that LightSplat achieves state-of-the-art performance with 50-400× faster feature distillation and 64× lower memory usage, offering a scalable and practical foundation for real-time 3D applications.

Acknowledgements

This work was supported by Institute of Information & communications Technology Planning & Evaluation (IITP) grant funded by the Korea government (MSIT) (No.RS-2020-II201336, Artificial Intelligence Graduate School Program (UNIST); No.RS-2025-25442824, AI Star Fellowship Program (UNIST); No.RS-2025-25442149, LG AI STAR Talent Development Program for Leading Large-Scale Generative AI Models in the Physical AI Domain), and by the InnoCORE program of the Ministry of Science and ICT (N10250156).

References

[1] Yash Bhalgat, Iro Laina, João F Henriques, Andrew Zisserman, and Andrea Vedaldi. N2f2: Hierarchical scene understanding with nested neural feature fields. In *European Conference on Computer Vision*, 2024. 1, 2, 13

[2] 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

[3] Yiwen Chen, Zilong Chen, Chi Zhang, Feng Wang, Xiaofeng Yang, Yikai Wang, Zhongang Cai, Lei Yang, Huaping Liu, and Guosheng Lin. Gaussianeditor: Swift and controllable 3d editing with gaussian splatting. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2024. 2

[4] Mehdi Cherti, Romain Beaumont, Ross Wightman, Mitchell Wortsman, Gabriel Ilharco, Cade Gordon, Christoph Schuhmann, Ludwig Schmidt, and Jenia Jitsev. Reproducible scaling laws for contrastive language-image learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2023. 11

[5] Angela Dai, Angel X Chang, Manolis Savva, Maciej Halber, Thomas Funkhouser, and Matthias Nießner. Scannet: Richly-annotated 3d reconstructions of indoor scenes. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2017. 2, 5

[6] Stephan J Garbin, Marek Kowalski, Matthew Johnson, Jamie Shotton, and Julien Valentin. Fastnerf: High-fidelity neural rendering at 200fps. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2021. 2

[7] Jun Guo, Xiaojian Ma, Yue Fan, Huaping Liu, and Qing Li. Semantic gaussians: Open-vocabulary scene understanding with 3d gaussian splatting. *arXiv preprint arXiv:2403.15624*, 2024. 1

[8] Mazeyu Ji, Ri-Zhao Qiu, Xueyan Zou, and Xiaolong Wang. Graspsplats: Efficient manipulation with 3d feature splatting. *arXiv preprint arXiv:2409.02084*, 2024. 1

[9] Ying Jiang, Chang Yu, Tianyi Xie, Xuan Li, Yutao Feng, Huamin Wang, Minchen Li, Henry Lau, Feng Gao, Yin Yang, et al. Vr-gs: A physical dynamics-aware interactive gaussian splatting system in virtual reality. In *ACM SIGGRAPH 2024 Conference Papers*, 2024. 2

[10] Kim Jun-Seong, GeonU Kim, Kim Yu-Ji, Yu-Chiang Frank Wang, Jaesung Choe, and Tae-Hyun Oh. Dr. splat: Directly referring 3d gaussian splatting via direct language embedding registration. In *Proceedings of the Computer Vision and Pattern Recognition Conference*, 2025. 1, 2, 3, 5, 13

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

[12] 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*, 2023. 1, 2

[13] Alexander Kirillov, Eric Mintun, Nikhila Ravi, Hanzi Mao, Chloe Rolland, Laura Gustafson, Tete Xiao, Spencer Whitehead, Alexander C Berg, Wan-Yen Lo, et al. Segment anything. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2023. 1, 2

[14] Sosuke Kobayashi, Eiichi Matsumoto, and Vincent Sitzmann. Decomposing nerf for editing via feature field distillation. *Advances in neural information processing systems*, 35:23311–23330, 2022. 2

[15] Siyun Liang, Sen Wang, Kunyi Li, Michael Niemeyer, Stefano Gasperini, Nassir Navab, and Federico Tombari. Supergseg: Open-vocabulary 3d segmentation with structured super-gaussians. *arXiv preprint arXiv:2412.10231*, 2024. 2

[16] Lu Ling, Yichen Sheng, Zhi Tu, Wentian Zhao, Cheng Xin, Kun Wan, Lantao Yu, Qianyu Guo, Zixun Yu, Yawen Lu, et al. Dl3dv-10k: A large-scale scene dataset for deep learning-based 3d vision. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2024. 2, 5

[17] Juliette Marrie, Romain Ménégaux, Michael Arbel, Diane Larlus, and Julien Mairal. Ludvig: Learning-free uplifting of 2d visual features to gaussian splatting scenes. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2025. 1, 3, 5, 13

[18] Ben Mildenhall, Pratul P Srinivasan, Matthew Tancik, Jonathan T Barron, Ravi Ramamoorthi, and Ren Ng. Nerf: Representing scenes as neural radiance fields for view synthesis. *Communications of the ACM*, 65(1):99–106, 2021. 1

[19] Thomas Neff, Pascal Stadlbauer, Mathias Parger, Andreas Kurz, Joerg H Mueller, Chakravarty R Alla Chaitanya, Anton Kaplanyan, and Markus Steinberger. Donerf: Towards real-time rendering of compact neural radiance fields using depth oracle networks. *Computer Graphics Forum*, 40(4):45–59, 2021. 2

[20] 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*, 2024. 1, 2, 3, 5, 13

[21] Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, et al. Learning transferable visual models from natural language supervision. In *International Conference on Machine Learning*, 2021. 1, 2

[22] 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*, 2024. 1, 2, 5, 13

[23] Nikolaos Tsagkas, Oisin Mac Aodha, and Chris Xiaoxuan Lu. Vl-fields: Towards language-grounded neural implicit spatial representations. *arXiv preprint arXiv:2305.12427*, 2023. 2

[24] Vadim Tschernezki, Iro Laina, Diane Larlus, and Andrea Vedaldi. Neural feature fusion fields: 3d distillation of self-supervised 2d image representations. In *2022 International Conference on 3D Vision (3DV)*, 2022. 2

[25] Peng Wang, Yuan Liu, Zhaoxi Chen, Lingjie Liu, Ziwei Liu, Taku Komura, Christian Theobalt, and Wenping Wang. F2-nerf: Fast neural radiance field training with free camera trajectories. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2023. 2

[26] Jing Wu, Jia-Wang Bian, Xinghui Li, Guangrun Wang, Ian Reid, Philip Torr, and Victor Adrian Prisacariu. Gaussctrl: Multi-view consistent text-driven 3d gaussian splatting editing. In *European Conference on Computer Vision*, 2024. 2

[27] Yanmin Wu, Jiarui Meng, Haijie Li, Chenming Wu, Yahao Shi, Xinhua Cheng, Chen Zhao, Haocheng Feng, Errui Ding, Jingdong Wang, et al. Opengaussian: Towards point-level 3d gaussian-based open vocabulary understanding. *Advances in Neural Information Processing Systems*, 37:19114–19138, 2024. 1, 2, 5, 11

[28] Qiangeng Xu, Zexiang Xu, Julien Philip, Sai Bi, Zhixin Shu, Kalyan Sunkavalli, and Ulrich Neumann. Point-nerf: Point-based neural radiance fields. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2022. 2

[29] Han Yan, Celong Liu, Chao Ma, and Xing Mei. Plenvdb: Memory efficient vdb-based radiance fields for fast training and rendering. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2023. 2

[30] Heng Yu, Joel Julin, Zoltán Á Milacski, Koichiro Niinuma, and László A Jeni. Cogs: Controllable gaussian splatting. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2024. 2

[31] Wenbo Zhang, Lu Zhang, Ping Hu, Liqian Ma, Yunzhi Zhuge, and Huchuan Lu. Bootstraping clustering of gaussians for view-consistent 3d scene understanding. In *Proceedings of the AAAI Conference on Artificial Intelligence*, 2025. 2

⚡ LightSplat: Fast and Memory-Efficient Open-Vocabulary 3D Scene Understanding in Five Seconds
Supplementary Materials

Overview

In these supplementary materials, we provide additional details on the following topics:

- Sec. A introduces the DL3DV-OVS dataset, a large-scale open-vocabulary benchmark covering four indoor-outdoor scenes with diverse object queries.
- Sec. B outlines our experimental setup, including 3DGS, feature extraction, and hyperparameters.
- Sec. C summarizes feature distillation time and reports millisecond-level inference from lightweight clustering.
- Sec. D presents additional qualitative comparisons that demonstrate robustness in diverse 3D scenes.
- Sec. E showcases text-driven 3D editing enabled by cluster-level semantic control.
- Sec. F discusses limitations regarding object feature selection and the precision-speed trade-off.

Table S1. **Scene-wise text queries in the DL3DV-OVS dataset.** Each scene provides a set of text queries for evaluating open-vocabulary 3D object selection.

Scene	Text Queries		
Park	bench	trash bin	car
	parking signpost	information board	
Shop	black drawer	framed artwork	lamp
	potted plant	white sofa	bed
	fire extinguisher	round wooden coffee table	
Road	bench	car	fire hydrant
	planter	roof	trash bin
	streetlight		
Office	chair	keyboard	monitor
	whiteboard	mouse	

A. DL3DV-OVS Dataset

We propose DL3DV-OVS, an open-vocabulary extension of DL3DV for evaluating model robustness in large-scale and challenging scenes. The dataset spans indoor and outdoor environments with broad spatial layouts and diverse scene structures. It provides 22 text queries for objects such as cars, chairs, monitors, and streetlights. These items vary significantly in size and distance, often appearing multiple times within a single view. DL3DV-OVS consists of four scenes: park, road, office, and shop. For each scene, we provide 960×540 multi-view images, text queries, and the corresponding ground-truth 2D masks. Table S1 lists the text queries for each scene.

B. More Implementation Details

B.1. Experimental Setup

Training Setup. For consistent comparison of speed and accuracy, all models are evaluated on a single RTX 4090 GPU. LUDVIG is the only exception, running on an A6000 because its high-dimensional language feature operations exceed the 24GB memory limit.

3D Gaussian Splatting. To ensure a fair comparison, we use a pre-trained 3DGS trained for 30,000 iterations, identical to LangSplat and OpenGaussian. We do not modify any Gaussian parameters and simply allocate an additional 2-byte space per Gaussian to store an index. A full language feature requires 2028 bytes per Gaussian, so storing it for 100,000 Gaussians would require about 200 MB, which is much larger than the 23.6 MB needed for all standard Gaussian parameters. In our method, we no longer store this 2024-byte feature for each Gaussian. Instead, each Gaussian stores only a small index, with the total index size being just 0.2 MB, which is less than 0.1% of the 200 MB required for per-Gaussian language features.

Object Feature Extraction. LangSplat uses all three hierarchical mask levels produced by SAM and learns separate language features for each level. To ensure both fair comparison and computational efficiency, we follow OpenGaussian and adopt a large-mask strategy. We extract mask-level CLIP features by cropping each SAM mask, resizing it to 224 × 224 with zero padding, and encoding it using OpenCLIP [4]. With this setup, our method integrates semantics and geometry without relying on such hierarchical structures, achieving SOTA performance with 50× faster speed.

B.2. Hyperparameters

To ensure fair evaluation, we use the same hyperparameters for all scenes within each dataset. Denser views or more compact scenes produce higher mask overlap and thus require stricter IoU thresholds. In contrast, datasets with more diverse text queries require slightly relaxed feature similarity thresholds. Accordingly, for LERF-OVS, ScanNet, and DL3DV-OVS, we set (contrib, noise, IoU, feat) to (0.04, 200, 0.6, 0.75), (0.04, 500, 0.35, 0.8), and (0.09, 450, 0.5, 0.8), respectively. Unlike prior methods that tune hyperparameters per scene (e.g., teatime) [27], our method uses fixed settings and still performs consistently well across diverse scenes, including the challenging ramen

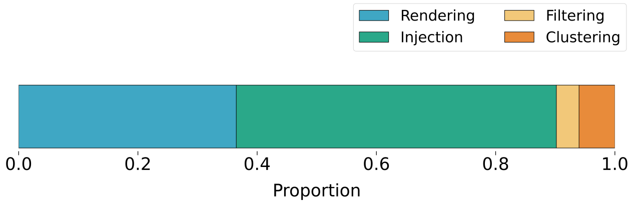

Figure S1. **Module-wise feature distillation time.** This figure shows the time proportion of each module during feature distillation. Injection, filtering, and clustering refer to indexed feature injection, 3D-aware mask filtering, and context-aware 3D clustering, respectively.

scene. Across all three datasets, our method achieves a 400-720× speedup in feature distillation and improves mIoU over OpenGaussian by 5.43, 7.68, and 19.65 points, respectively. These results demonstrate strong robustness across diverse environments.

B.3. Evaluation Setup

Model Comparison. Since Dr.Splat provides only training code, we adopt the reported inference metrics from its paper and measure all other results ourselves. Dr.Splat is expected to have higher inference time, as it processes 128-dimensional features through a codebook and compares them against all Gaussians. In contrast, our method performs semantic comparison at the 3D cluster level instead of evaluating every individual Gaussian. By attaching semantics to clusters through an index-feature mapping without relying on a codebook, the comparison set remains compact and efficient. As shown in Table 3, this enables our method to achieve a 2ms inference time on average across ScanNet scenes.

3D Object Selection. This section provides a detailed explanation of the 3D Object Selection task. Previous works, such as LangSplat and LEGaussians, struggle to accurately distinguish objects in 3D space. This is because these methods distill features after rendering 3D Gaussians into 2D, where overlapping Gaussians blur the features and result in indirect, less effective supervision. As a result, high-quality features extracted in 2D often do not transfer well to 3D, and 2D-based evaluation provides only a partial view of the model's 3D understanding. To address this, we adopt the 3D Object Selection task, which directly selects objects in 3D from text queries. We evaluate on LERF-OVS by first selecting objects directly in 3D, then rendering them back to 2D for comparison with ground-truth masks, following OpenGaussian and Dr.Splat.

3D Semantic Segmentation. We evaluate 3D semantic segmentation on the ScanNet benchmark, following the evaluation protocol used in OpenGaussian. The protocol fits 3D Gaussians directly to the ScanNet ground-truth points, matching their locations and counts. This establishes a one-to-one correspondence between Gaussians and GT points, enabling direct comparison of predicted and ground-truth labels. For fair comparison, all baselines follow the same evaluation setting.

Table S2. **Inference time comparison on LERF-OVS and DL3DV-OVS.** We report inference time in seconds. While prior methods rely on slow 2D or per-Gaussian processing, `LightSplat`'s cluster-level inference achieves millisecond-level runtime.

Methods	LERF-OVS	DL3DV-OVS
LangSplat	1.046	0.175
LEGaussians	0.383	0.770
LUDVIG	5.923	1.865
OpenGaussian	0.003	0.005
Ours	0.001	0.003

C. More Quantitative Results

Module FD Time. Fig. S1 shows the average time consumed by each module in our feature distillation pipeline across the LERF-OVS scenes. Most of the computational cost arises from rendering and indexed feature injection, since both steps require computation for each view. In contrast, the subsequent 3D-aware mask filtering and context-aware clustering operate in a single step, adding almost no additional cost or delay. This design keeps the overall distillation process extremely fast, and the ablation study shows that these components improve performance without slowing the pipeline. Across scenes, rendering, injection, filtering, and clustering account for 36.5%, 53.7%, 3.8%, and 6.0% of the total time, respectively.

Inference Time. Table S2 presents inference-time results on LERF-OVS and DL3DV-OVS, providing additional evaluations beyond the ScanNet results included in the main paper. Rendering-based methods such as LangSplat and LEGaussians incur significant computational overhead because their inference pipeline requires additional decoding of semantic features. LangSplat performs per-Gaussian latent decoding through its autoencoder, while LEGaussians applies per-pixel semantic decoding using its MLP, both of which further increase latency. OpenGaussian is also slow because it computes high-dimensional features for every Gaussian. LUDVIG introduces additional overhead by diffusing 2048-dimensional features over a graph. In contrast, `LightSplat` performs cluster-level inference through an index-feature mapping, avoiding per-Gaussian comparisons and diffusion. As a result, `LightSplat` maintains millisecond-level inference even in large indoor-outdoor scenes.

D. More Qualitative Results

We present additional qualitative results omitted from the main paper due to space constraints. These results demonstrate the robustness of our model across a wide range of scenes.

Clustering Process. Fig. S2 illustrates how `LightSplat` converts 2D masks into stable 3D object clusters. We use the same hyperparameters for the two DL3DV-OVS scenes shown above and the two additional indoor-outdoor scenes below, demonstrating robustness without scene-specific tuning. The process begins with SAM masks extracted from the input images. These masks are injected into the 3D scene based on per-pixel contribution, producing a mask ID field where each Gaussian receives the index of its most influential 2D mask. This intermediate field may appear noisy due to view-dependent inconsistencies. We address this by filtering out unstable masks in the 3D-aware mask filtering stage using 2D-3D correspondences. Finally, context-aware 3D clustering groups semantically and spatially consistent masks into clean object-level clusters. The resulting cluster ID field demonstrates robust and coherent semantics across diverse indoor and outdoor environments.

3D Object Selection. Fig. S3 and Fig. S4 present additional qualitative results on the LERF-OVS and DL3DV-OVS datasets. Unlike models where iterative aggregation blurs semantics in 3D [1, 10, 17, 20, 22], our method transfers mask-level semantics directly into 3D. It then merges related masks into object-level clusters, preserving much clearer object structure than Gaussian-level aggregation. As a result, our method achieves high accuracy in challenging real-world cases:

- small objects attached to others (kamaboko, wavy noodle)
- multiple instances of the same object (knives, cars)
- semantically subtle categories (yellow pouf)
- thin or complex structures (jake with thin legs, streetlight)
- single outdoor objects (trash bin, information board)

3D Semantic Segmentation. Fig. S5 presents more qualitative results on the ScanNet dataset. These results show that our approach captures a broader range of semantics in indoor scenes, while also producing more complete object coverage with well-aligned boundaries.

These additional qualitative results further highlight the robustness of our model to diverse and challenging queries, from small objects to complex structures, while outperforming other models in both accuracy and speed on LERF-OVS, ScanNet, and DL3DV-OVS.

E. 3D Scene Editing

Fig. S6 shows how our method naturally extends to 3D scene editing. By managing semantics at the cluster level, our method enables fast and accurate segmentation of text-specified objects in 3D space. Building on this segmentation capability, users can directly remove, recolor, or reposition the identified objects, enabling high-fidelity scene edits. We demonstrate this capability through recoloring and enlarging as representative examples. Enlargement is performed by adjusting Gaussian distances and scales, while recoloring is achieved by modifying SH coefficients. Notably, these editing operations add minimal overhead, keeping inference time nearly unchanged in interactive scenarios. Together, these results highlight the speed, accuracy, and flexibility of our approach for interactive 3D manipulation in immersive content creation, AR/VR, and robotics.

F. Limitations

Object Feature Selection. SAM provides high-quality object segmentation masks, and CLIP enables effective image-text interaction as a foundation model trained on large-scale datasets in a shared embedding space. However, SAM masks do not always produce perfectly accurate boundaries. In addition, CLIP can struggle to distinguish between semantically similar objects (e.g., red apple vs. green apple) due to their high feature similarity. These limitations could be further mitigated with extra modules or iterative training, but this would disrupt the speed–accuracy balance of our approach.

Hyperparameters. Our method performs semantic injection and 3D clustering in a single step using only four main hyperparameters, making it lightweight and intuitive. Compared to training-based methods, which involve numerous hyperparameters for model initialization and optimization, our approach requires far fewer parameters, avoiding the high training cost and memory usage. Despite its simplicity, our method can still be influenced by the choice of hyperparameters and can be further refined for improvement.

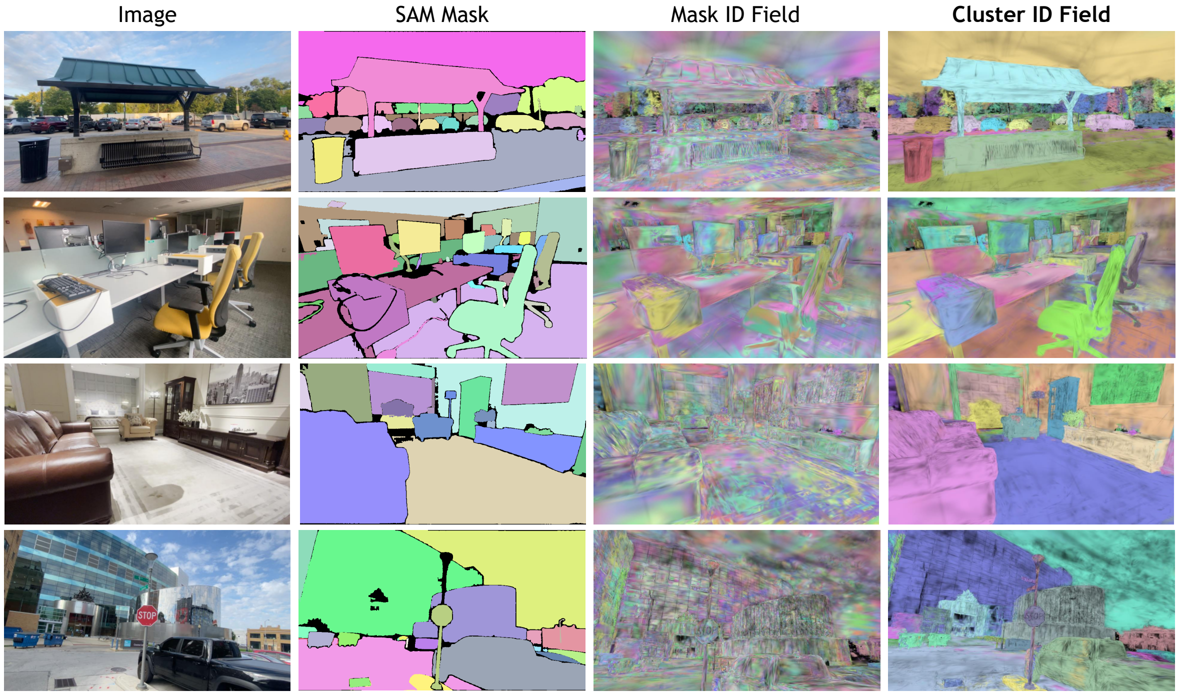

Figure S2. **Clustering Process Visualization.** The figure illustrates how SAM mask IDs are lifted into 3D and refined into coherent object-level clusters, resulting in the final cluster ID field. It demonstrates that our method produces clean cluster ID fields across diverse indoor and outdoor scenes without scene-specific tuning.

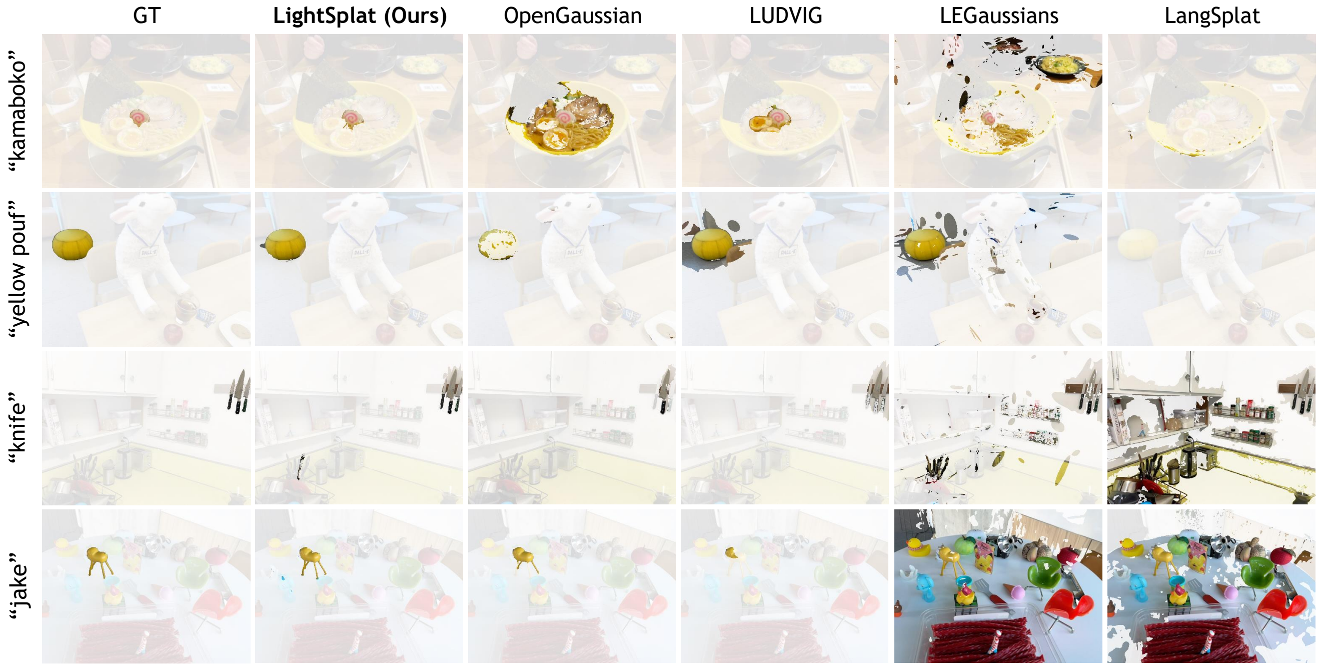

Figure S3. **More qualitative comparison for 3D Object Selection on the LERF-OVS dataset.** We visualize model performance across different scenes and text queries in LERF-OVS. With context-aware 3D clustering, our method delivers precise boundaries for challenging queries, including small objects in contact with others, repeated instances, and thin or intricate structures, while achieving the fastest performance.

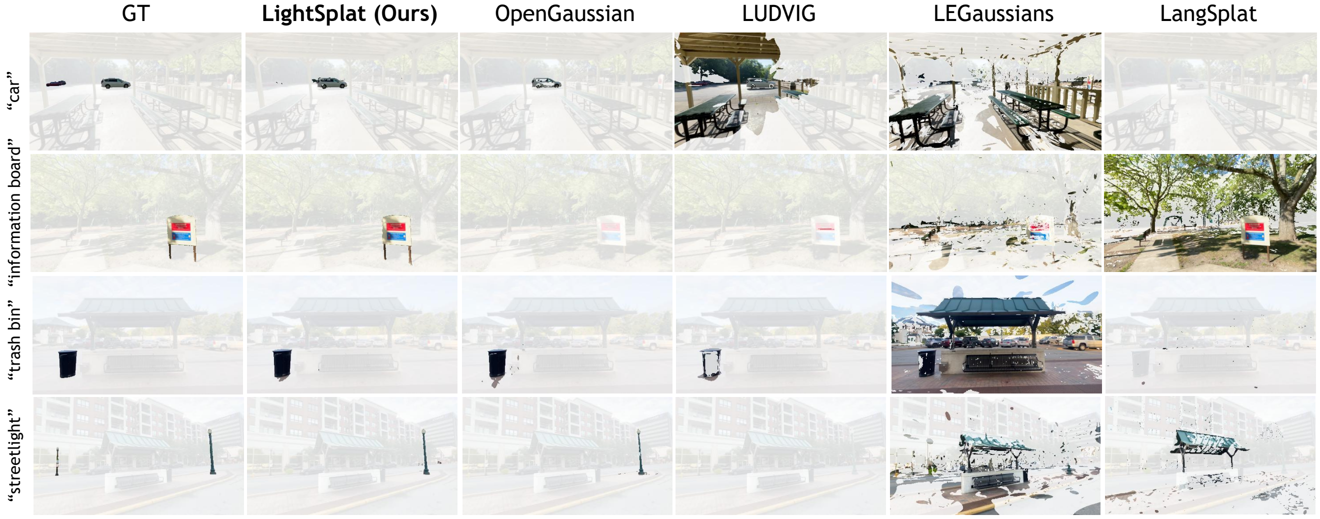

Figure S4. **More qualitative comparison for 3D Object Selection on the DL3DV-OVS dataset.** We visualize model performance across different scenes and text queries in DL3DV-OVS. With context-aware 3D clustering, our method produces accurate boundaries in large indoor-outdoor scenes, handling distant objects, multiple instances, and complex structures while maintaining the fastest performance.

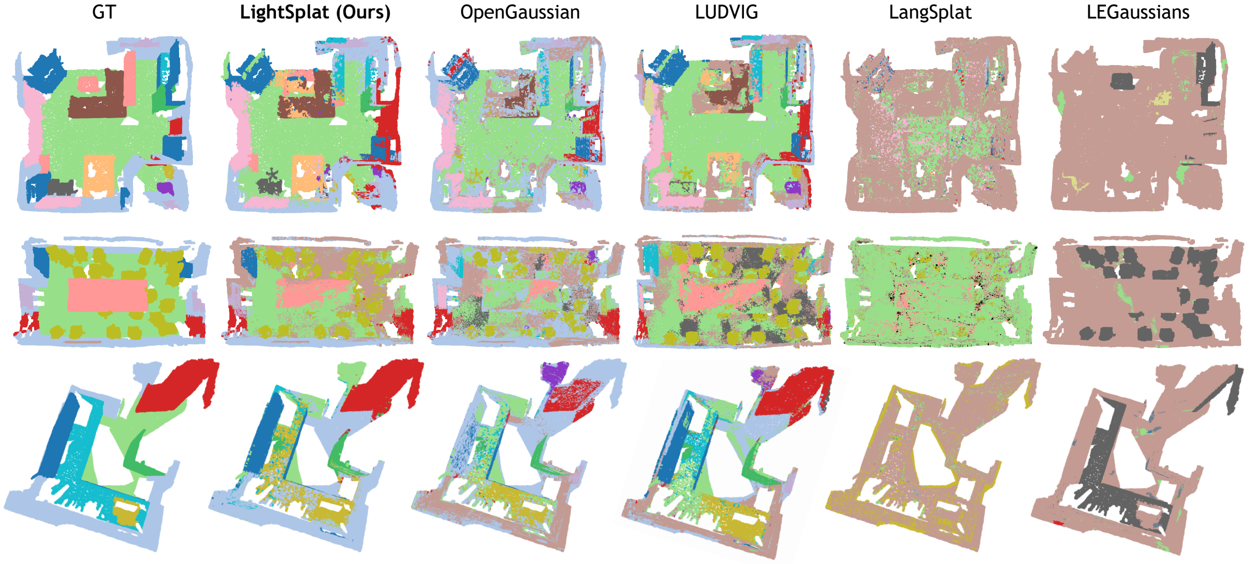

Figure S5. **More qualitative comparison for 3D Semantic Segmentation on the ScanNet dataset.** For intuitive comparison, ground-truth semantics are visualized in distinct colors. Our method captures a broader range of semantics in indoor scenes, providing more complete coverage with precise boundaries. It handles both object-level (e.g., door, cabinet) and large-area semantics (e.g., floor, wall), showing robustness across diverse scenes and outperforming other methods in accuracy and speed.

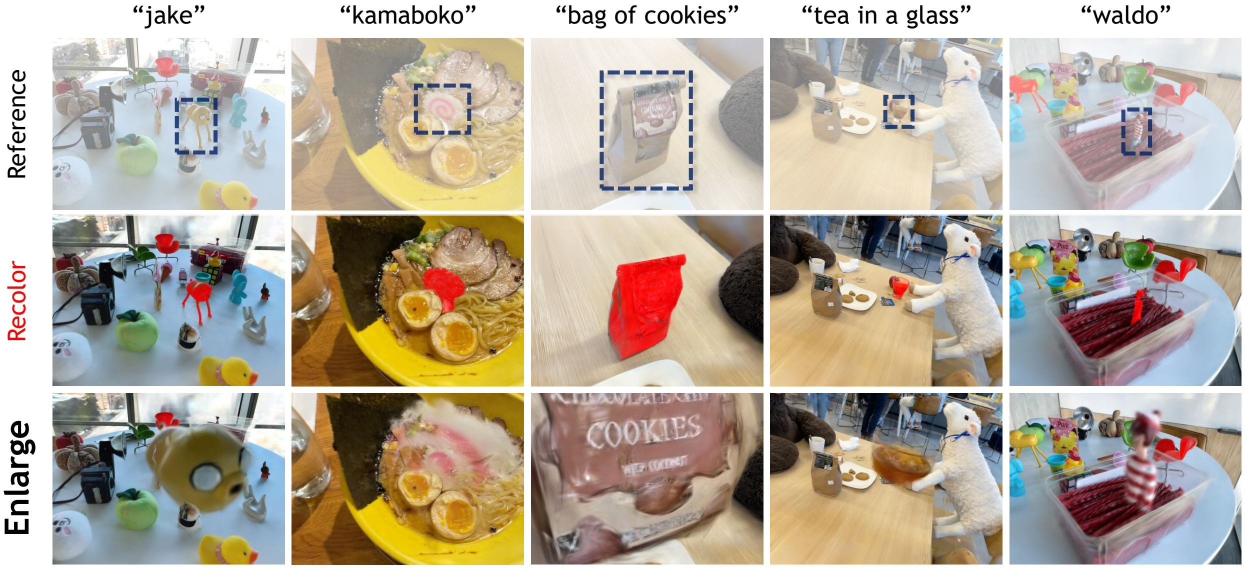

Figure S6. **Qualitative results of text-driven 3D scene editing on the LERF-OVS dataset.** Recoloring (middle) and enlarging (bottom) are shown as representative examples, enabled by our 3DGS-based approach for fast, accurate, and high-fidelity object manipulation across diverse scenes.