CCL-LGS: Contrastive Codebook Learning for 3D Language Gaussian Splatting

Lei Tian[1], Xiaomin Li[1], Liqian Ma[2], Hefei Huang[1], Zirui Zheng[1],

Hao Yin[1], Taiqing Li[1], Huchuan Lu[1], Xu Jia[1,✉]

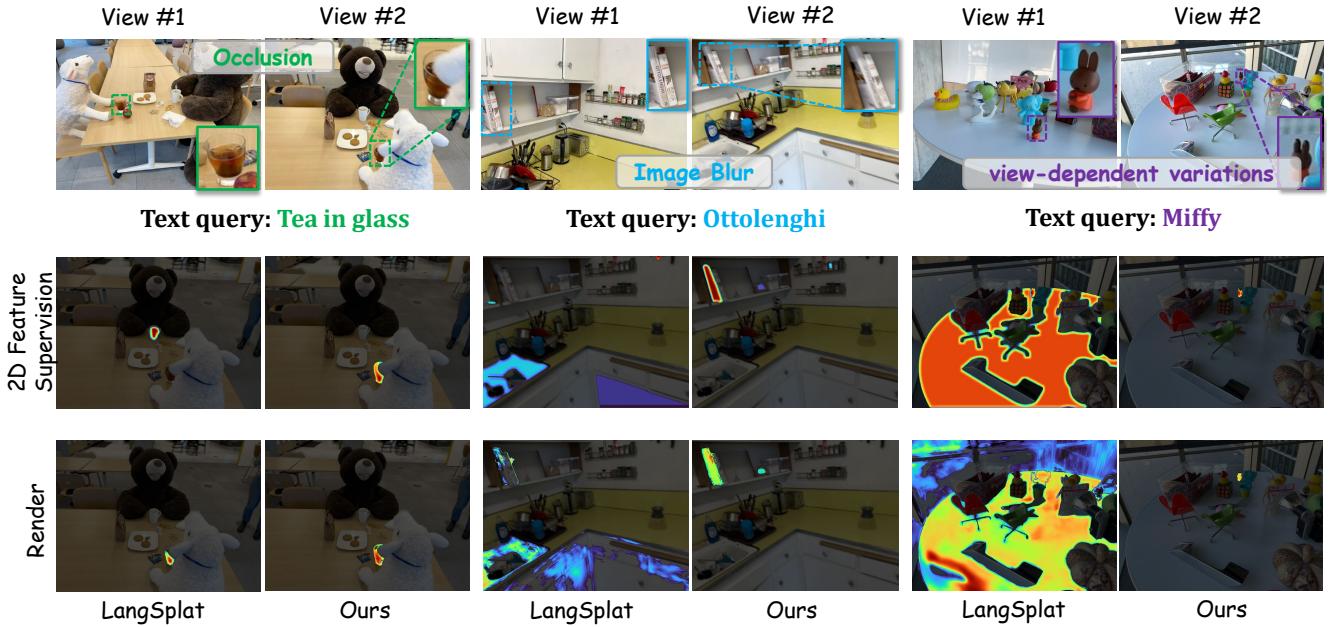[1]Dalian University of Technology   [2]ZMO AI

Figure 1. Quantitative comparison of our method and LangSplat under three challenging scenarios: Occlusion, Image Blur, and View-Dependent Variations. The results clearly demonstrate the superior performance of our approach, which exhibits greater robustness and fidelity in handling these challenges compared to LangSplat.

Abstract

Recent advances in 3D reconstruction techniques and vision-language models have fueled significant progress in 3D semantic understanding, a capability critical to robotics, autonomous driving, and virtual/augmented reality. However, methods that rely on 2D priors are prone to a critical challenge: cross-view semantic inconsistencies induced by occlusion, image blur, and view-dependent variations. These inconsistencies, when propagated via projection supervision, deteriorate the quality of 3D Gaussian semantic fields and introduce artifacts in the rendered outputs. To mitigate this limitation, we propose CCL-LGS, a novel framework that enforces view-consistent semantic supervision by integrating multi-view semantic cues. Specifically, our approach first employs a zero-shot tracker to align a set of SAM-generated 2D masks and reliably identify their corresponding categories. Next, we utilize CLIP to extract robust semantic encodings across views. Finally, our Contrastive Codebook Learning (CCL) module distills discriminative semantic features by enforcing intra-class compactness and inter-class distinctiveness. In contrast to previous methods that directly apply CLIP to imperfect masks, our framework explicitly resolves semantic conflicts while preserving category discriminability. Extensive experiments demonstrate that CCL-LGS outperforms previous state-of-the-art methods. Our project page is available at `https://epsilontl.github.io/CCL-LGS/`.

1. Introduction

Significant progress has been made in 3D reconstruction, particularly with the advent of 3D Gaussian splatting(3DGS) [7], which enables the generation of high-

---

✉Corresponding authors.

fidelity 3D color representations and supports real-time rendering of novel viewpoints. Meanwhile, vision-language models such as CLIP [23] and LSeg [14] have bridged the gap between the two modalities, enabling the generation of rich and dense semantic maps for images without the need for additional supervision. With such advancing technologies, 3D semantic understanding, which aims to obtain 3D semantic representations from multi-view images and corresponding camera poses, has made rapid progress. This task has a wide range of applications, including robotics [29], autonomous driving [9], and VR/AR [21].

This task is particularly challenging due to factors such as semantic ambiguity (e.g., a point on the tip of the nose may correspond to both "nose" and "face" queries) and the long-tail distribution of vocabulary queries. Early methods like LERF [8] tackle these issues by extending NeRF [16] with multi-scale CLIP features and dynamic 2D mapping selection, achieving accurate semantic localization. However, their reliance on exhaustive multi-scale rendering leads to inefficiency, and patch-based feature extraction often fails to capture precise object boundaries, resulting scale misalignment and performance degradation. Building upon the rise of explicit 3D representations such as 3DGS [7], recent works [18, 20, 22] differ from earlier NeRF-based methods by leveraging visual foundation models to extract dense, pixel-level semantic features that guide the construction of 3D semantic fields. While differing in implementation, these methods generally follow a common paradigm: supervising 3D semantic representations by projecting them to 2D views and comparing the rendered results with features extracted from pre-trained vision-language models.

However, this paradigm based on 2D priors relies heavily on the assumption that semantic supervision remains consistent across different views. In practice, as illustrated in Fig. 1, factors such as occlusion, motion blur, and view-dependent variations can introduce significant inconsistencies in the 2D semantic features of the same object across viewpoints. Recent methods [18, 20, 22, 25] built upon 3DGS [7] reconstruct a 3D semantic field from 2D features and leverage 3D geometric consistency to partially address cross-view semantic inconsistencies. However, since they still rely on 2D supervision, significant inconsistencies in the input features can propagate into the 3D space. This makes it difficult to maintain semantic coherence across views and often leads to artifacts in the rendered novel views. While many existing approaches enforce geometric constraints to enhance multi-view consistency, they often overlook the explicit modeling of 2D feature alignment across views—an underexplored yet critical factor that limits semantic reconstruction performance in open-world scenarios.

In this paper, we propose CCL-LGS, a novel framework for view-consistent 3D semantic reconstruction. Our key

innovation lies in establishing view-consistent semantic supervision through a specially designed three-stage pipeline, enabling the reconstruction of a 3D Gaussian semantic field. Specifically, we first extract accurate instance masks using SAM [11], then align cross-view correspondences via zero-shot tracking, and finally distill semantics through a Contrastive Codebook Learning (CCL) module. The proposed CCL module introduces contrastive metric learning to enforce intra-class feature compactness while maintaining inter-class feature distinctiveness. This design effectively mitigates semantic ambiguities introduced by incomplete or noisy masks. Unlike prior approaches that directly apply CLIP to imperfect masks, our framework not only establishes reliable semantic correspondences across views but also preserves category-wise distinctiveness, leading to more robust and consistent 3D semantic reconstruction. Owing to its proficiency in 3D open-vocabulary scene understanding, our method could benefit a variety of downstream applications. The main contributions of our work can be summarized as follows:

- We propose a novel framework, CCL-LGS, which integrates view-consistent semantic supervision to enable the reconstruction of 3D Gaussian semantic fields.
- We develop a Contrastive Codebook Learning (CCL) module that resolves semantic ambiguities by enforcing intra-class compactness and inter-class distinctiveness, enabling robust semantic representation even with noisy or incomplete masks.
- Extensive experiments on benchmark datasets demonstrate that our approach achieves state-of-the-art performance in open-vocabulary semantic segmentation tasks.

2. Related Work

3D Neural Representations. Recent advances in 3D scene representation have yielded impressive results, particularly with neural radiation fields (NeRF) [16], which has excelled in novel view synthesis by generating highly realistic new perspectives. However, NeRF's reliance on implicit neural networks to model entire scenes results in lengthy training and rendering times. In contrast, explicit and hybrid scene representation approaches [3, 4, 19, 28, 30] typically employ techniques such as hash grids and point clouds to alleviate the computational burden associated with large neural networks. Notably, 3D Gaussian Splatting (3DGS) [7] has emerged as a promising alternative by representing scenes as collections of 3D Gaussian ellipsoids that can be efficiently rasterized into images. This method not only accelerates the rendering process but also allows for intuitive manipulation of individual scene components. Motivated by the success of 3D Gaussian Splatting in novel view synthesis, many studies have extended its application to other domains [1, 6, 13, 26, 31] to fully leverage its efficient rendering capabilities. In this paper, we utilize 3D Gaussian

2

Splatting [7] for 3D neural representations.

Visual Foundation Models. Recent advances in visual foundation models have redefined computer vision by leveraging massive datasets and high-capacity architectures to tackle a wide range of tasks. Models such as CLIP [23] employ contrastive learning to integrate visual and textual modalities, while the Segment Anything Model (SAM) [11] delivers impressive zero-shot performance by generating high-quality segmentation masks across diverse scenarios. Self-supervised models like DINO [33] and DINOv2 [17] further enrich this landscape by extracting fine-grained semantic features, including object boundaries and scene layouts, which are critical for downstream applications. In parallel, advances in video object segmentation have opened new avenues for extending these robust visual representations into the temporal domain. Recent video segmentation methods focus on maintaining consistent object masks across frames, thus capturing dynamic temporal cues. For instance, XMem [2] leverages zero-shot tracking strategies, employing a memory mechanism to robustly segment objects in videos despite challenges like occlusion or appearance variations. SAM [11] is an image segmentation model designed for single images, while SAM2 [24] builds upon its capabilities to enable video object segmentation. By taking a video sequence and reference frame masks as input, SAM2 aims to address the challenge of long-term tracking by consistently predicting object masks in any target frame across extended video sequences. Therefore, in this paper, we employ SAM and CLIP to extract ground-truth features for baseline comparisons, using SAM2 for mask matching, ensuring a fair evaluation.

3D Scene Understanding. Earlier works, such as Semantic NeRF [34] and Panoptic NeRF [5], pioneered the integration of 2D semantic or panoptic annotations into 3D radiance field representations, enabling zero-shot scene understanding. Building on this foundation, subsequent studies [12, 27] explored the use of pixel-aligned semantic features directly lifted into 3D, thereby moving beyond reliance on predetermined semantic labels. More recent methods for scene understanding focus on developing multi-modal 3D representations from posed images, supporting novel viewpoint rendering for tasks such as open-vocabulary semantic segmentation. For NeRF-based techniques, LERF [8] integrates a language field into NeRF by leveraging spatial positions and physical scales to produce CLIP vectors. Among 3DGS-based methods, LangSplat [20] utilizes an autoencoder to perform dimensionality reduction on multi-scale features for extracting 2D semantic features. LEGaussians [25] and GOI [22] classify the extracted features using a codebook-based approach. Building upon the features extracted by LangSplat, 3D VL-GS [18] places greater emphasis on language features by incorporating data enrichment and a cross-modal rasterizer.

Although these methods adopt different technical strategies, they all share a reliance on projection-based supervision and overlook the issue of semantic inconsistency.

3. Method

In this section, we present our proposed framework, CCL-LGS, for view-consistent 3D semantic reconstruction. As illustrated in Fig. 2, we first extract two-level semantic features from multi-view images (Sec. 3.2), then perform mask association and contrastive codebook learning to organize and refine these features (Sec. 3.3), and finally integrate the semantic information into the 3D Gaussian semantic field (Sec. 3.4).

3.1. Preliminary: 3DGS with Language Features

3D Gaussian Splatting utilizes a set of 3D Gaussians, represented as Gaussian ellipsoids, to model a scene. Each 3D Gaussian is parameterized by its centroid $p \in \mathbb{R}^3$, its covariance, which is defined by a rotational quaternion $r \in \mathbb{R}^4$ and a scaling factor $s \in \mathbb{R}^3$. To enable fast α-blending for rendering, each Gaussian is also associated with an opacity value $\alpha \in \mathbb{R}$ and a color vector c, represented in the three degrees of spherical harmonics (SH) coefficients. To further enhance scene representation, Gaussian splatting models have been extended to incorporate dense language-embedding information [20, 25, 35]. This is achieved by introducing a low-dimensional (LD) language feature vector $f \in \mathbb{R}^{d_f}$ for each Gaussian. In summary, the learnable parameters of the i-th 3D Gaussian are represented as: $\theta_i = \{p_i, r_i, s_i, \alpha_i, c_i, f_i\}$.

Images are rasterized by splatting Gaussians onto each pixel v in the scene and performing α-blending to compute the final color $C(v)$, as defined by:

$$C(v) = \sum_{i \in \mathcal{N}} c_i \alpha_i \prod_{j=1}^{i-1}(1 - \alpha_j), \quad (1)$$

where $\mathcal{N}$ denotes the set of Gaussians contributing to the tile. Similarly, the predicted semantic feature at pixel v is rendered as:

$$\hat{F}(v) = \sum_{i \in \mathcal{N}} f_i \alpha_i \prod_{j=1}^{i-1}(1 - \alpha_j). \quad (2)$$

3.2. Two-Level Semantic Feature Extraction

Existing approaches for 3D semantic understanding that incorporate language embeddings [8, 20, 22, 25] have demonstrated promising performance. However, some methods [8, 25] rely on multi-scale patch averaging for pixel-level semantic feature extraction, which often leads to blurred boundaries. In contrast, GOI's single-scale mechanism struggles to resolve semantic ambiguities effectively.
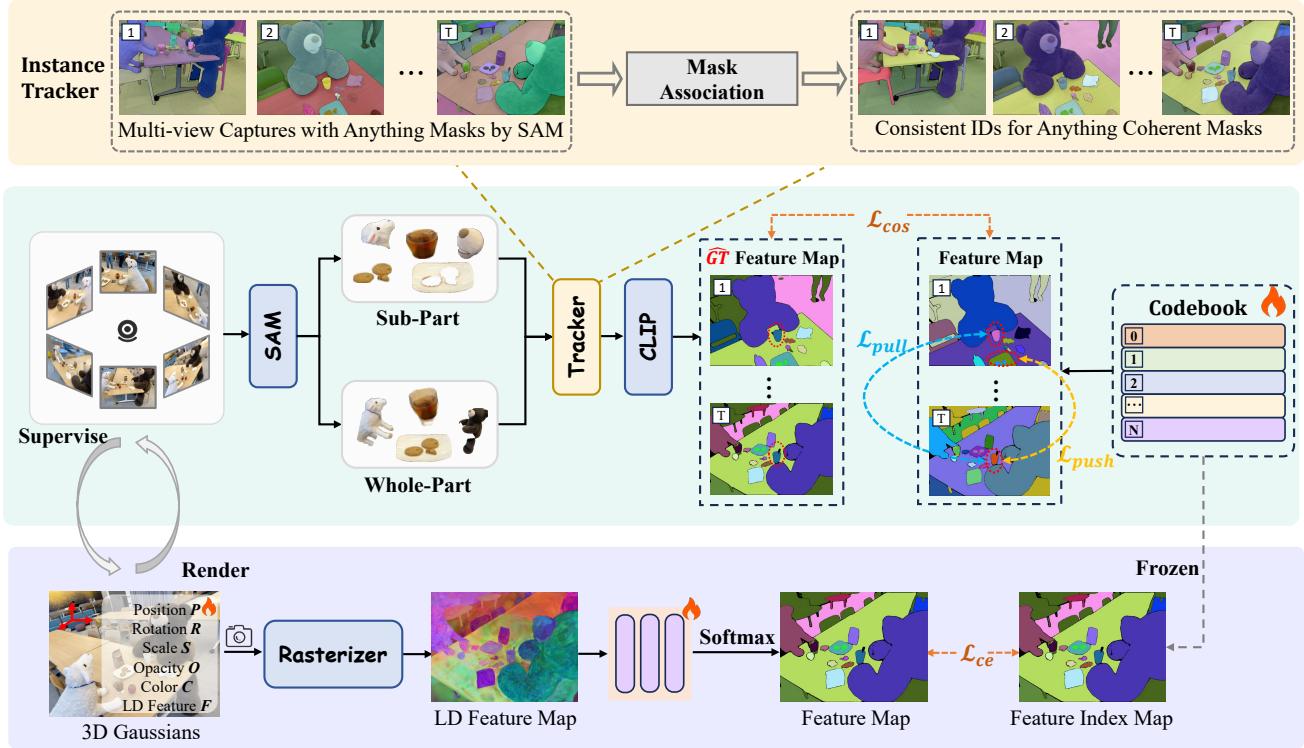
Figure 2. The framework of our CCL-LGS. Top: Instance tracker responsible for mask association. Middle: CCL module that constructs consistent 2D semantic supervision. For multi-view scene images, we first extract dual-scale masks, then perform mask association through the tracker and extract semantics using CLIP. The CCL module subsequently enhances intra-class compactness and inter-class distinctiveness in the categorized semantic features, effectively mitigating ambiguities caused by incomplete or noisy masks. Bottom: The optimization process of the 3D Gaussian semantic field. For each training view, low-dimensional (LD) features from 3D Gaussians are rendered into 2D maps. The optimization is driven by supervision using cross-entropy loss.

Although LangSplat [20] extracts object-level features with clear boundaries by generating masks for subparts, parts, and whole objects, its dependence on multiple models increases data processing and training times, and it may introduce scale errors due to CLIP ambiguities and potential information loss during training. To address these challenges, our approach harnesses SAM's robust segmentation capabilities to achieve precise boundaries while effectively consolidating multi-scale semantic information within a unified framework.

In our method, a uniform 32×32 point prompt is provided to SAM to generate three types of masks corresponding to the semantic scales of subparts, parts, and whole objects. Recognizing that different points may yield conflicting scale assignments (for example, a subpart for one point might be considered a part for another), we merge the subpart and part masks as well as the whole and part masks to obtain two aggregated sets of masks, denoted as M_o^{sp} and M_o^{wp}. A filtering procedure is then applied to eliminate redundant masks based on predicted IoU scores, stability met-

rics, and overlap rates, with the final non-overlapping masks M_i^l (where $l \in \{sp, wp\}$ and i indexes the masks) selected based on the product of the IoU and stability scores. For clarity, we omit the superscript l in all subsequent expressions (e.g., M_i instead of M_i^l), with the implicit understanding that operations are independently applied to each scale l. After obtaining segmentation maps, pixel-aligned language embeddings can be extracted using CLIP. For each pixel v, its semantic feature $F_i(v)$ can be expressed as:

$$F_i(v) = \text{CLIP}(I_t \odot M_i(v)), \quad (3)$$

where I_t is the input image, and $M_i(v)$ denotes the mask region to which pixel v belongs.

The design of this module is motivated by the need to balance computational efficiency with multi-scale semantic precision. By integrating mask generation and feature extraction within a unified framework, our approach reduces computational overhead while ensuring high semantic accuracy and precise boundary delineation.

4

3.3. Contrastive Codebook Learning

Having obtained a set of high-dimensional features, recent methods [18, 20, 22, 25] typically use an autoencoder or codebook to obtain low-dimensional semantic features. These low-dimensional features are then used to supervise the low-dimensional semantic encoding stored in 3D Gaussians. However, due to occlusion, image blur, and view-dependent variations, applying CLIP directly to imperfect masks results in inconsistent semantic features, ultimately affecting the quality of 3D semantic field reconstruction.

To mitigate the limitations of directly using features derived from imperfect masks, we introduce a codebook-based contrastive learning approach. This approach consists of two key steps: (1) mask association via IoU matching and (2) applying contrastive losses to improve feature representation.

The first step corresponds to the "Mask Association" module at the top of Fig. 2. Specifically, we first propagate the K masks from the first frame to all frames using SAM2 [24]. For the t-th frame, the propagated K masks are compared with all masks M_i in that frame using IoU. If the maximum IoU between a propagated mask and a mask M_i exceeds 0.5, then M_i is assigned the category of the matching propagated mask; otherwise, it is assigned the category -1 to indicate an unmatched mask. Hence, each mask M_i is assigned a label $y_i \in \{1, 2, \ldots, K, -1\}$.

In the second step, a codebook $T = \{T_j\}_{j=1}^N$ is constructed, where each prototype $T_j \in \mathbb{R}^d$ is learned during training. Note that N is independent of the $K + 1$ mask categories. Specifically, N represents a fixed capacity for scene-specific feature learning, while K refers to the number of object categories observed in the current scene subset, which may be incomplete due to limited observations. The codebook serves as latent feature prototypes that structure the feature space for contrastive learning. Contrastive losses are then applied to encourage the alignment of features with the same category (pull loss) and the separation of features with different categories (push loss), as shown in the middle of Fig. 2.

For each feature F_i, we search for the most similar prototype in T based on cosine similarity:

$$j^* = \underset{j}{\text{argmax}} \ \cos(F_i, T_j), \qquad (4)$$

where $\cos(F_i, T_j)$ denotes the cosine similarity between the feature F_i and the prototype T_j. To encourage F_i to match well with its closest prototype, we define a matching loss:

$$L_{\text{max}} = 1 - \cos(F_i, T_{j^*}). \qquad (5)$$

Contrastive losses are then applied based on the assigned mask labels. For two features F_i and F_k with corresponding mask labels y_i and y_k:

If $y_i = y_k$ and $y_i \neq -1$, we apply a pull loss to pull their associated descriptors closer via their codebook projections:

$$L_{\text{pull}} = 1 - \cos(T_{j_i}, T_{j_k}), \qquad (6)$$

where T_{j_i} and T_{j_k} are the prototypes selected for F_i and F_k, respectively.

If $y_i \neq y_k$ and $y_i, y_k \neq -1$, we apply a push loss to force their codebook representations apart:

$$L_{\text{push}} = \text{ReLU}(\cos(T_{j_i}, T_{j_k}) - m), \qquad (7)$$

where $m > 0$ is a predefined margin.

For features with $y_i = -1$ (unmatched masks), neither pull nor push loss is applied. Finally, the total loss is formulated as a weighted sum:

$$L = L_{\text{max}} + \lambda_{\text{pull}} L_{\text{pull}} + \lambda_{\text{push}} L_{\text{push}}, \qquad (8)$$

with weighting factors λ_{pull}, and λ_{push} controlling the contribution of each component.

This framework design has two advantages. First, compared to autoencoders, similar features in the feature space are implicitly constrained to the same table entry, resulting in stronger consistency constraints. Second, through contrastive learning losses, the framework ensures that features corresponding to the same mask category become well clustered, while those from different mask categories are effectively separated. This improves the semantic consistency of imperfect masks and increases the distinction between semantics, resulting in better 2D supervised representations.

3.4. 3D Gaussian Semantic Field

Given a trained codebook $T = \{T_j\}_{j=1}^N$, we construct the 3D Gaussian semantic field as illustrated at the bottom of Fig. 2, by converting per-pixel semantic features into discrete indices and aligning these indices with the outputs of 3D Gaussian Splatting.

Specifically, for each pixel v, its semantic feature $F_i(v)$ is obtained via Eq. (3). The index j^* is assigned to v through Eq. (4), generating a semantic index map $\mathcal{M} \in \mathbb{R}^{H \times W}$, where H and W denote the image height and width, respectively. Subsequently, a feature map $\hat{F} \in \mathbb{R}^{H \times W \times d_f}$ is rendered via rasterization and alpha blending (see Sec. 3.1). This feature map is processed by a lightweight MLP decoder followed by a softmax layer to produce a semantic feature distribution $\hat{\mathcal{M}} \in \mathbb{R}^{H \times W \times N}$, where N corresponds to the number of codebook categories. To jointly optimize the semantic features of 3D Gaussians and the parameters of the MLP decoder, we minimize the cross-entropy loss:

$$\mathcal{L}_{\text{CE}} = \text{CE}(\hat{\mathcal{M}}, \mathcal{M}), \qquad (9)$$

At inference, each pixel retrieves its prototype $T_{\hat{\mathcal{M}}(v)}$ to form the refined semantic feature $\tilde{F}(v)$. Given a text query

τ, we compute its embedding $\varphi(\tau)$ via the text encoder of the vision-language model to compute the relevance map.

$$p(\tau \mid v) = \frac{\exp\left(\frac{\tilde{F}(v)\cdot\varphi(\tau)}{\|\tilde{F}(v)\|\|\varphi(\tau)\|}\right)}{\sum_{s\in\mathcal{T}}\exp\left(\frac{\tilde{F}(v)\cdot\varphi(s)}{\|\tilde{F}(v)\|\|\varphi(s)\|}\right)}. \quad (10)$$

Thresholding $p(\tau \mid v)$ yields a semantic segmentation mask for the queried concept.

4. Experiments

Dataset. To assess the effectiveness of our approach, we conduct experiments on two benchmark datasets using the mean Intersection over Union (mIoU) metric. The first dataset, LERF [8], is captured using the Polycam application on an iPhone and features four challenging indoor scenes: Ramen, Figurines, Teatime, and Waldo Kitchen. These scenes are annotated with pixel-accurate ground truth masks for textual queries, as provided by the LangSplat [20]. The dataset's real-world imaging conditions, including severe occlusions and motion blur, make it particularly suited for testing segmentation robustness in complex environments. The second dataset, 3D-OVS [15], consists of long-tail objects set against diverse backgrounds. For our evaluation, we focus on four specific scenes: Bed, Bench, Lawn, and Sofa. Note that the Room scene contains a significant annotation error; thus, we exclude it from quantitative evaluation and provide qualitative results only in the supplementary material.

Implementation Details. To extract semantic features of each image, we utilize the SAM ViT-H model [11] alongside the OpenCLIP ViT-B/16 model [23]. For the contrastive loss coefficients, we set $\lambda_{\text{pull}} = \lambda_{\text{push}} = 0.25$ and use a margin m of 0.7 to ensure adequate separation between feature clusters. For each scene, we jointly train the 3D Gaussians for both color and semantic features by setting $d_c = 3$ and $d_f = 8$. The training is performed over 30,000 iterations using the Adam optimizer [10], with a learning rate of 0.001 and beta parameters set to $(0.9, 0.999)$.

Baseline. For a fair comparison, we select the latest works on open-vocabulary 3D scene understanding: Feature-3DGS[35], LEGaussians[25], LangSplat[20], GS-Grouping[32], GOI[22], and 3D VL-GS[18].

4.1. Experiments on LERF

Quantitative Results. We first compare our method with existing SOTA methods on LERF dataset. As shown in Tab. 1. We observed that our method achieved an IoU result of 65.6 in 3D semantic segmentation, ranking either first or second across all four scenes, outperforming the state-of-the-art 3D Vision-Language GS by 3.6. This illustrates the superiority of our proposed CCL-LGS method.

Method	Ramen	Figurines	Teatime	Waldo Kitchen	Avg.
Feature-3DGS	43.7	40.5	58.8	39.6	45.7
LEGaussians	46.0	40.8	60.3	39.4	46.6
LangSplat	51.2	44.7	65.1	44.5	51.4
GS-Grouping	45.5	40.0	60.9	38.7	46.3
GOI	52.6	44.5	63.7	41.4	50.6
3D vl-gs	<u>61.4</u>	58.1	**73.5**	54.8	<u>62.0</u>
Ours	**62.3**	**61.2**	<u>71.8</u>	**67.1**	**65.6**

Table 1. Quantitative experiments results on LERF dataset. The best result is bolded, and the second-best result is underlined.

Method	Ramen	Figurines	Teatime	Waldo Kitchen	Avg.
Baseline	46.8	57.1	60.8	61.0	56.4
baseline(w/ pull loss)	48.0	58.0	70.1	62.0	59.5
baseline(w/ push loss)	55.1	61.0	66.0	59.3	60.4
Ours	**62.3**	**61.2**	**71.8**	**67.1**	**65.6**

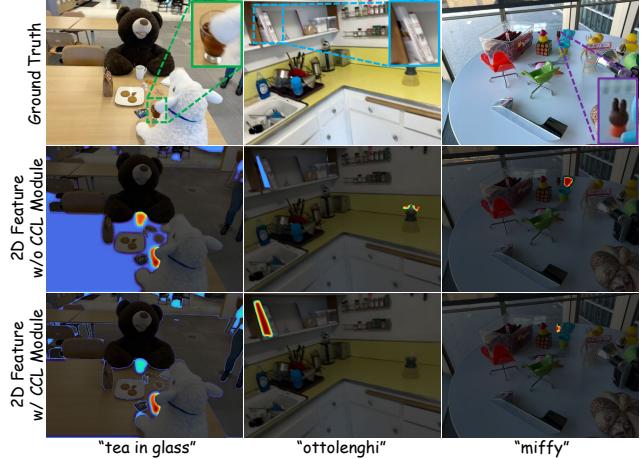Table 2. Ablation study on LERF dataset.

Figure 3. Qualitative comparison of 2D feature maps with and without CCL module.

Visualization Results. Fig. 4 illustrates segmentation results for two scenes: figurines (top) and kitchen (bottom). In the figurines scene, we compare how each method segments the same object across two different viewpoints, revealing that competing approaches often exhibit inconsistent segmentations. In contrast, our method provides stable and accurate results, even under viewpoint changes. In the kitchen scene, we specifically focus on the cabinet, a challenging object that other methods frequently fail to segment correctly. Our CCL-LGS framework effectively captures its boundaries, demonstrating both superior cross-view consistency and strong performance on difficult categories.

Ablation study. To validate the effectiveness of the proposed Contrastive Codebook Learning (CCL) module, we conduct a series of experiments, including visual analysis of 2D supervision features and ablation studies on 3D semantic segmentation. As shown in Fig. 3, features refined by the CCL module exhibit enhanced spatial precision and
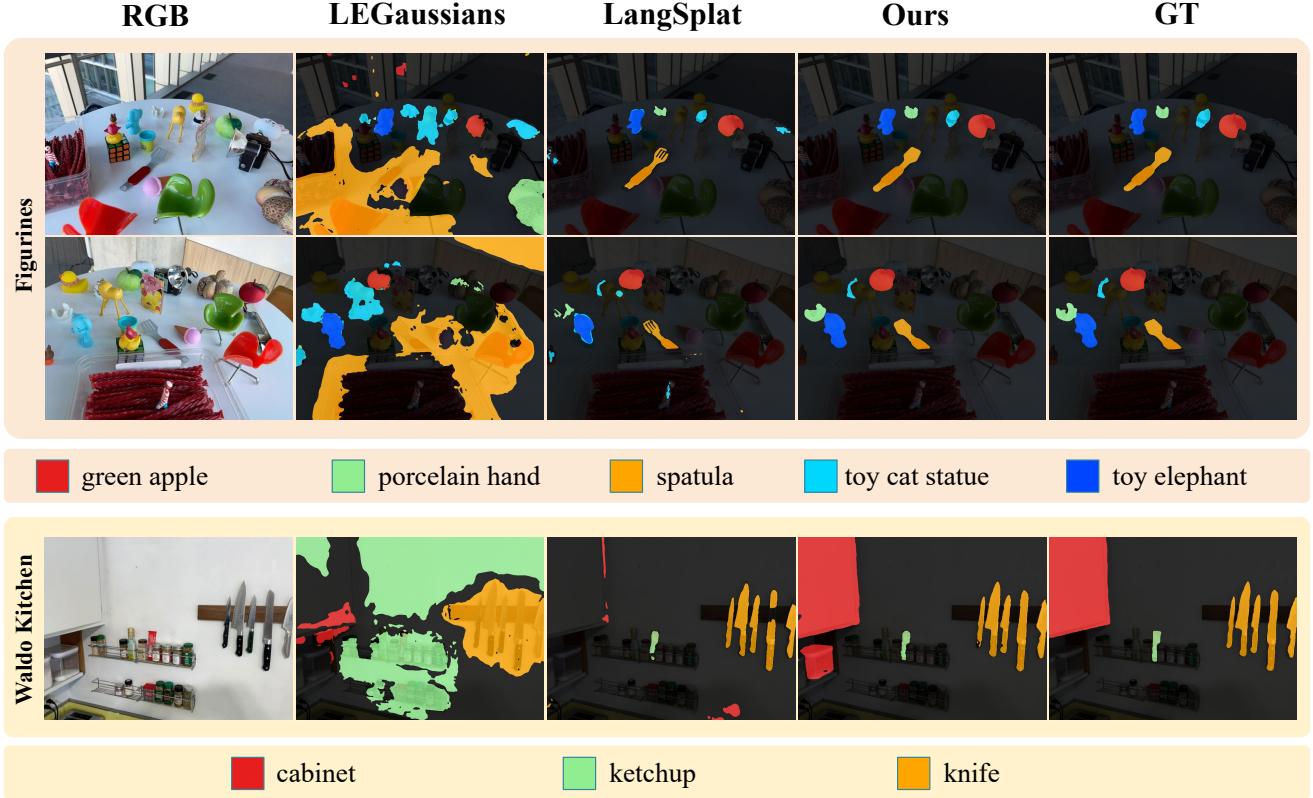
6

Figure 4. Segmentation results on the figurines (top) and kitchen (bottom) scenes. Our method achieves consistent multi-view segmentation and accurately captures challenging objects like the cabinet, outperforming prior approaches.
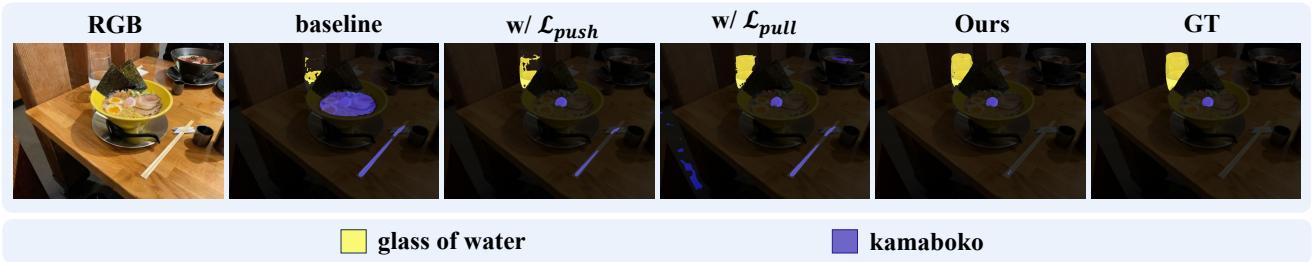

Figure 5. Qualitative comparison of different loss configurations. The pull loss improves intra-class consistency (e.g., for "glass of water"), while the push loss reduces false activations (e.g., around the "kamaboko"). The full model effectively combines both.

stronger activation on semantically relevant regions, eliminating the scattered and ambiguous responses observed in the baseline. This visual evidence demonstrates that CCL improves the alignment between extracted features and semantic ground truth during 2D supervision generation.

We further perform ablation studies to quantitatively analyze the impact of individual loss components on 3D semantic understanding. Evaluations are conducted on four scenes from the LERF dataset. The baseline model uses codebook-based feature compression to produce 2D semantic supervision for 3D segmentation. We compare three variants: baseline with pull loss, baseline with push loss, and the full model with both losses. As shown in Tab. 2, both losses are essential for optimal performance—removing either leads to noticeable degradation, though all variants still outperform the baseline.

Qualitative results in Fig. 5 further reveal the complementary roles of the two losses. The pull loss enhances intra-class consistency, particularly for occluded or partially visible objects like the "glass of water", while the push loss suppresses false activations in confusing regions, such as around the "kamaboko". The full model combines both effects to ensure robust and discriminative 3D semantic segmentation across challenging scenes.
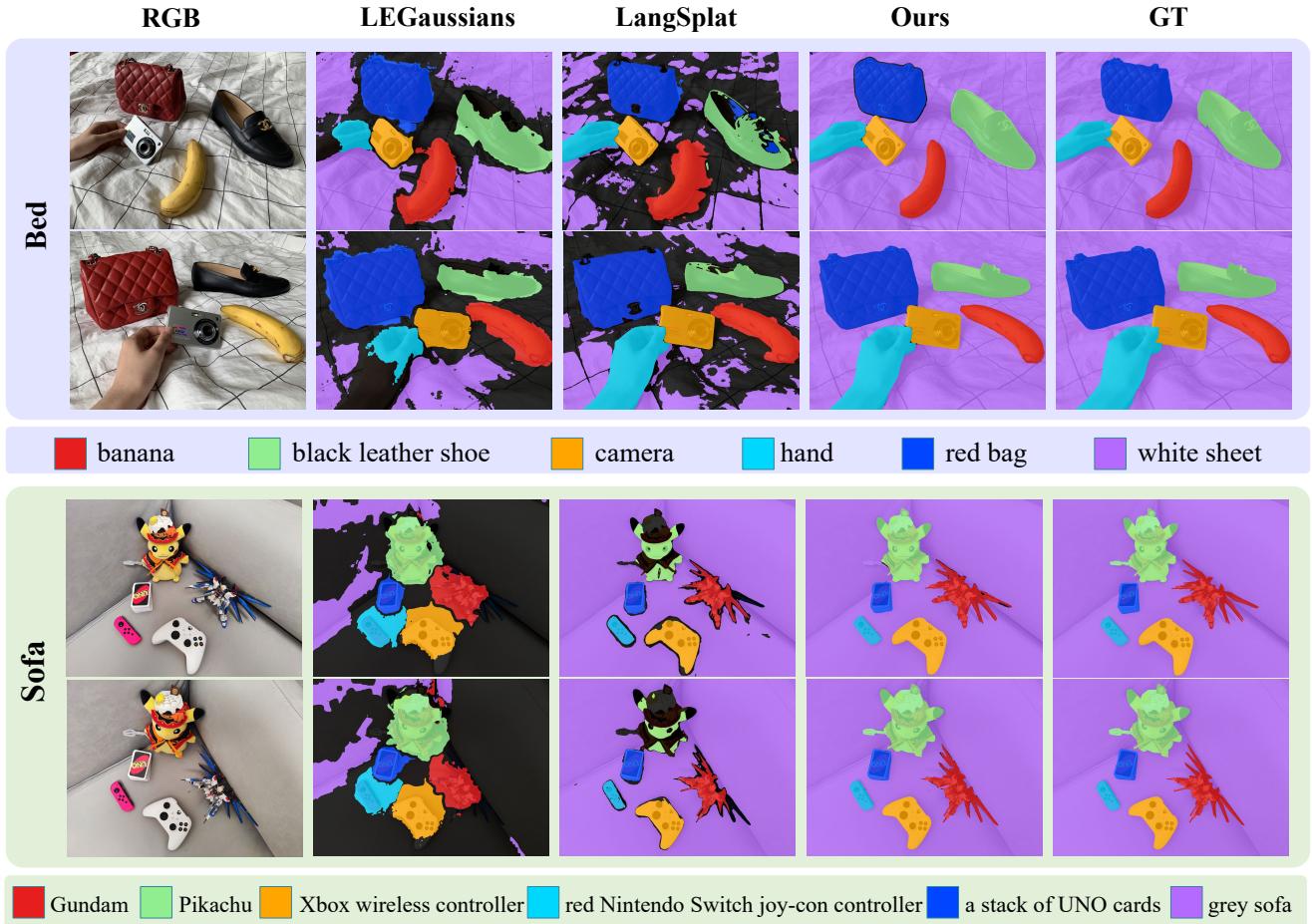
7

Figure 6. Qualitative comparison on 3D-OVS dataset. In these two scenes, our method clearly outperforms the other two methods.

Method	Bed	Bench	Sofa	Lawn	Avg.
Feature-3DGS	83.5	90.7	86.9	93.4	88.6
LEGaussians	84.9	91.1	87.8	92.5	89.1
LangSplat	92.5	94.2	90.0	<u>96.1</u>	93.2
GS-Grouping	83.0	91.5	87.3	90.6	88.1
GOI	89.4	92.8	85.6	94.1	90.5
3D VL-GS	<u>96.8</u>	**97.3**	**95.5**	**97.9**	**96.9**
Ours	**97.3**	<u>95.0</u>	<u>92.3</u>	96.1	<u>95.2</u>

Table 3. Quantitative experiments results on 3D-OVS dataset. The best result is bolded, and the second-best result is underlined.

4.2. Experiments on 3D-OVS

Quantitative Results. We compare our method with existing state-of-the-art approaches on the 3D-OVS dataset, as shown in Tab. 3. While our approach achieves comparable performance, it underperforms 3D VL-GS. We attribute this to the nature of the 3D-OVS dataset, which is relatively small and simple, with minimal occlusion and blur—conditions under which the data enrichment strategy employed by 3D VL-GS offers more noticeable benefits.

Qualitative Results. We present our qualitative experimental results in Figure 6. Since LEGaussians is a patch-based method, it learns over-smoothed features and fails to capture sharp object boundaries. LangSplat adopts a three-scale architecture but does not enhance multi-view features as our method does. In our experiments, we observed that it sometimes selects inappropriate scales for certain objects, leading to suboptimal segmentation performance. Among the compared methods, our method produces the most accurate segmentation maps, further demonstrating the effectiveness of our CCL-LGS.

5. Conclusion

In this paper, we introduced CCL-LGS, a novel method for constructing 3D language fields that enables accurate and efficient open-vocabulary queries in 3D spaces. Our work tackles a previously overlooked challenge: applying CLIP directly to imperfect masks produces inconsistent semantic features, and using these features to supervise 3D genera-

tion can lead to artifacts. To address this issue, we proposed a dedicated CCL module that constructs compact and distinct features, thereby providing more reliable supervision. Extensive experiments demonstrate that our approach achieves state-of-the-art performance. Our method is limited by the capabilities of SAM and CLIP, with performance affected by imperfect instance masks, and future work will focus on improving mask refinement for greater robustness.

References

[1] Zilong Chen, Feng Wang, Yikai Wang, and Huaping Liu. Text-to-3d using gaussian splatting. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 21401–21412, 2024. 2

[2] Ho Kei Cheng and Alexander G Schwing. Xmem: Long-term video object segmentation with an atkinson-shiffrin memory model. In *European Conference on Computer Vision*, pages 640–658. Springer, 2022. 3

[3] Forrester Cole, Kyle Genova, Avneesh Sud, Daniel Vlasic, and Zhoutong Zhang. Differentiable surface rendering via non-differentiable sampling. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 6088–6097, 2021. 2

[4] Peng Dai, Yinda Zhang, Xin Yu, Xiaoyang Lyu, and Xiaojuan Qi. Hybrid neural rendering for large-scale scenes with motion blur. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 154–164, 2023. 2

[5] Xiao Fu, Shangzhan Zhang, Tianrun Chen, Yichong Lu, Lanyun Zhu, Xiaowei Zhou, Andreas Geiger, and Yiyi Liao. Panoptic nerf: 3d-to-2d label transfer for panoptic urban scene segmentation. In *2022 International Conference on 3D Vision (3DV)*, pages 1–11. IEEE, 2022. 3

[6] Quankai Gao, Qiangeng Xu, Zhe Cao, Ben Mildenhall, Wenchao Ma, Le Chen, Danhang Tang, and Ulrich Neumann. Gaussianflow: Splatting gaussian dynamics for 4d content creation. *arXiv preprint arXiv:2403.12365*, 2024. 2

[7] 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, 2, 3

[8] Justin Kerr, Chung Min Kim, Ken Goldberg, Angjoo Kanazawa, and Matthew Tancik. Lerf: Language embedded radiance fields. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 19729–19739, 2023. 2, 3, 6

[9] Mustafa Khan, Hamidreza Fazlali, Dhruv Sharma, Tongtong Cao, Dongfeng Bai, Yuan Ren, and Bingbing Liu. Autosplat: Constrained gaussian splatting for autonomous driving scene reconstruction. *arXiv preprint arXiv:2407.02598*, 2024. 2

[10] DiederikP. Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *arXiv: Learning,arXiv: Learning*, 2014. 6

[11] 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 any-

[12] 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. 3

[13] Jiahui Lei, Yufu Wang, Georgios Pavlakos, Lingjie Liu, and Kostas Daniilidis. Gart: Gaussian articulated template models. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 19876–19887, 2024. 2

[14] Boyi Li, Kilian Q Weinberger, Serge Belongie, Vladlen Koltun, and René Ranftl. Language-driven semantic segmentation. *arXiv preprint arXiv:2201.03546*, 2022. 2

[15] Kunhao Liu, Fangneng Zhan, Jiahui Zhang, Muyu Xu, Yingchen Yu, Abdulmotaleb El Saddik, Christian Theobalt, Eric Xing, and Shijian Lu. Weakly supervised 3d openvocabulary segmentation, 2023. 6

[16] 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. 2

[17] Maxime Oquab, Timothée Darcet, Théo Moutakanni, Huy Vo, Marc Szafraniec, Vasil Khalidov, Pierre Fernandez, Daniel Haziza, Francisco Massa, Alaaeldin El-Nouby, et al. Dinov2: Learning robust visual features without supervision. *arXiv preprint arXiv:2304.07193*, 2023. 3

[18] Qucheng Peng, Benjamin Planche, Zhongpai Gao, Meng Zheng, Anwesa Choudhuri, Terrence Chen, Chen Chen, and Ziyan Wu. 3d vision-language gaussian splatting. *arXiv preprint arXiv:2410.07577*, 2024. 2, 3, 5, 6

[19] Sergey Prokudin, Qianli Ma, Maxime Raafat, Julien Valentin, and Siyu Tang. Dynamic point fields. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 7964–7976, 2023. 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*, pages 20051–20060, 2024. 2, 3, 4, 5, 6

[21] Shi Qiu, Binzhu Xie, Qixuan Liu, and Pheng-Ann Heng. Advancing extended reality with 3d gaussian splatting: Innovations and prospects. In *2025 IEEE International Conference on Artificial Intelligence and eXtended and Virtual Reality (AIxVR)*, pages 203–208. IEEE, 2025. 2

[22] Yansong Qu, Shaohui Dai, Xinyang Li, Jianghang Lin, Liujuan Cao, Shengchuan Zhang, and Rongrong Ji. Goi: Find 3d gaussians of interest with an optimizable open-vocabulary semantic-space hyperplane. In *Proceedings of the 32nd ACM International Conference on Multimedia*, pages 5328–5337, 2024. 2, 3, 5, 6

[23] 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 supervi-

thing. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 4015–4026, 2023. 2, 3, 6

sion. In *International conference on machine learning*, pages 8748–8763. PmLR, 2021. 2, 3, 6

[24] Nikhila Ravi, Valentin Gabeur, Yuan-Ting Hu, Ronghang Hu, Chaitanya Ryali, Tengyu Ma, Haitham Khedr, Roman Rädle, Chloe Rolland, Laura Gustafson, et al. Sam 2: Segment anything in images and videos. *arXiv preprint arXiv:2408.00714*, 2024. 3, 5

[25] Jin-Chuan Shi, Miao Wang, Hao-Bin Duan, and Shao-Hua Guan. Language embedded 3d gaussians for open-vocabulary scene understanding. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 5333–5343, 2024. 2, 3, 5, 6

[26] Jiaxiang Tang, Jiawei Ren, Hang Zhou, Ziwei Liu, and Gang Zeng. Dreamgaussian: Generative gaussian splatting for efficient 3d content creation. *arXiv preprint arXiv:2309.16653*, 2023. 2

[27] 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)*, pages 443–453. IEEE, 2022. 3

[28] Yuze Wang, Junyi Wang, Yansong Qu, and Yue Qi. Rip-nerf: Learning rotation-invariant point-based neural radiance field for fine-grained editing and compositing. In *Proceedings of the 2023 ACM international conference on multimedia retrieval*, pages 125–134, 2023. 2

[29] Yuxuan Wu, Lei Pan, Wenhua Wu, Guangming Wang, Yanzi Miao, Fan Xu, and Hesheng Wang. Rl-gsbridge: 3d gaussian splatting based real2sim2real method for robotic manipulation learning. *arXiv preprint arXiv:2409.20291*, 2024. 2

[30] 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*, pages 5438–5448, 2022. 2

[31] Chi Yan, Delin Qu, Dan Xu, Bin Zhao, Zhigang Wang, Dong Wang, and Xuelong Li. Gs-slam: Dense visual slam with 3d gaussian splatting. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 19595–19604, 2024. 2

[32] Mingqiao Ye, Martin Danelljan, Fisher Yu, and Lei Ke. Gaussian grouping: Segment and edit anything in 3d scenes. In *European Conference on Computer Vision*, pages 162–179. Springer, 2024. 6

[33] Hao Zhang, Feng Li, Shilong Liu, Lei Zhang, Hang Su, Jun Zhu, Lionel M Ni, and Heung-Yeung Shum. Dino: Detr with improved denoising anchor boxes for end-to-end object detection. *arXiv preprint arXiv:2203.03605*, 2022. 3

[34] Shuaifeng Zhi, Tristan Laidlow, Stefan Leutenegger, and Andrew J Davison. In-place scene labelling and understanding with implicit scene representation. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 15838–15847, 2021. 3

[35] Shijie Zhou, Haoran Chang, Sicheng Jiang, Zhiwen Fan, Zehao Zhu, Dejia Xu, Pradyumna Chari, Suya You, Zhangyang Wang, and Achuta Kadambi. Feature 3dgs: Supercharging 3d gaussian splatting to enable distilled feature fields. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 21676–21685, 2024. 3, 6

CCL-LGS: Contrastive Codebook Learning for 3D Language Gaussian Splatting

Supplementary Material

A. Qualitative Results on the Room Scene

As mentioned in the main paper, the Room scene in the 3D-OVS dataset includes annotation errors specifically in the "wood wall" region. This mislabeling affects the reliability of quantitative evaluation. Therefore, we present qualitative results in Fig. 7.

B. Efficiency Analysis

We conducted additional experiments on the Ramen scene from the LERF dataset using an Intel i7-14700KF CPU and an NVIDIA RTX 4090 GPU. The results are summarized in Tab. 4. The rendering speed (FPS) is measured by rendering the images with language features at a consistent resolution. Results show that our method achieves a favorable balance between multi-scale segmentation accuracy and efficiency under constrained GPU memory conditions.

C. Scale Number Analysis

To explore the impact of scale granularity on semantic segmentation accuracy, we compare our two-scale design with a baseline that merges all masks from three scales (subparts, parts, and whole objects) into a single set, as shown in Tab. 5. This single-scale baseline simplifies processing but sacrifices scale-specific representation.

Although using all three separate scales may provide marginal performance gains, we found that it leads to prohibitive GPU memory usage, system memory consumption, and preprocessing time due to the overwhelming number of fine-grained masks generated at the subpart level. Given these limitations, we do not include full three-scale experiments.

Our method merges subpart with part masks and whole with part masks, producing two non-overlapping, semantically meaningful sets. This strategy reduces redundancy and computational overhead while preserving scale-aware distinctions.

D. Language-based 3D Interaction Capability

Although our main paper emphasizes 2D supervision for semantic learning, our method indeed supports direct 3D interaction via language queries. Our codebook-based approach allows users to perform language-based querying and editing directly in 3D space.

Specifically, given a language query, we first match the query embedding with the codebook to select relevant semantic categories. Then, without any rasterization or alpha blending, we directly classify the semantic features of each

Method	mIoU	Pre-process	Training	Total	FPS	Memory
LangSplat	51.2	256min	36min	292min	42	7GB+4GB
LEGaussians	46.0	2min	40min	42min	65	19GB+9GB
Ours	62.3	16min	74min	90min	65	11GB+9GB

Table 4. Efficiency comparison on the Ramen scene from the LERF dataset.

Method	Ramen	Figurines	Teatime	Waldo Kitchen	Avg.
single scale	44.3	57.6	70.1	57.7	57.4
ours	62.3	61.2	71.8	67.1	65.6

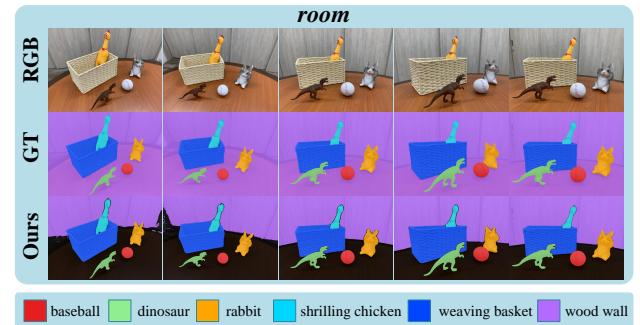Table 5. Comparison of different scale aggregation strategies.

Figure 7. Qualitative results on the Room scene. The "wood wall" category contains obvious annotation errors.
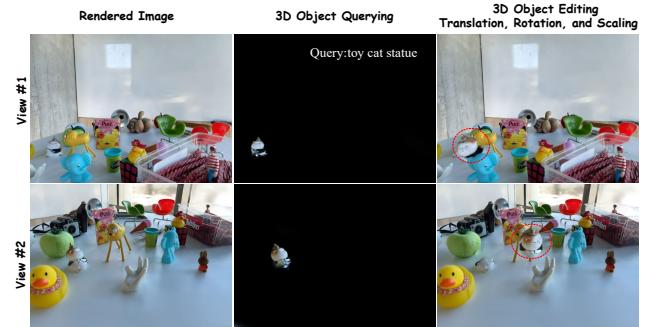

Figure 8. Examples of language-based 3D interaction and editing enabled by our method.

3D Gaussian using a lightweight linear classifier. Based on the predicted category distribution, we can identify Gaussians that correspond to the selected categories. This enables us to identify Gaussians in the 3D space that are semantically aligned with the input query. Once the relevant Gaussians are localized, we can directly perform various interaction operations on them, thereby enabling intuitive and interpretable 3D editing driven by natural language. Examples of such 3D interactions are illustrated in Fig. 8.

E. More Results

Beyond the specific cases discussed, we provide more qualitative visualizations of our method's semantic segmentation results across different scenes in Figs. 9, 10, and 11.
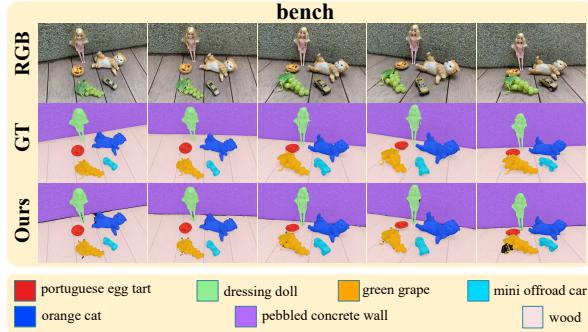

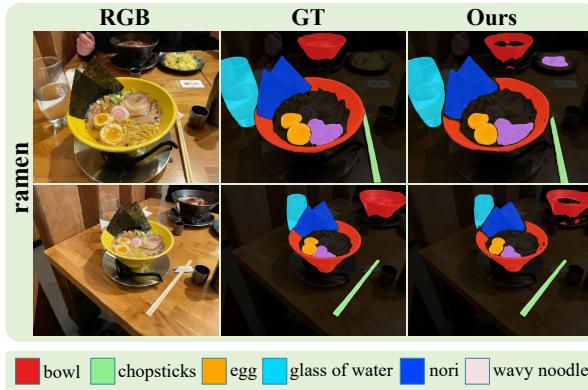Figure 9. Qualitative semantic segmentation results on the Bench scene.

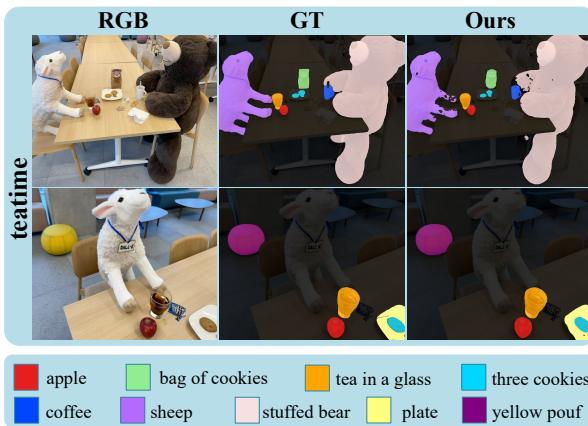Figure 10. Qualitative semantic segmentation results on the Ramen scene.

Figure 11. Qualitative semantic segmentation results on the Teatime scene.