

Lifting by Gaussians: A Simple, Fast and Flexible Method for 3D Instance Segmentation

Rohan Chacko, Nicolai Häni, Eldar Khaliullin, Douglas Lee, Lin Sun
Magic Leap Inc.

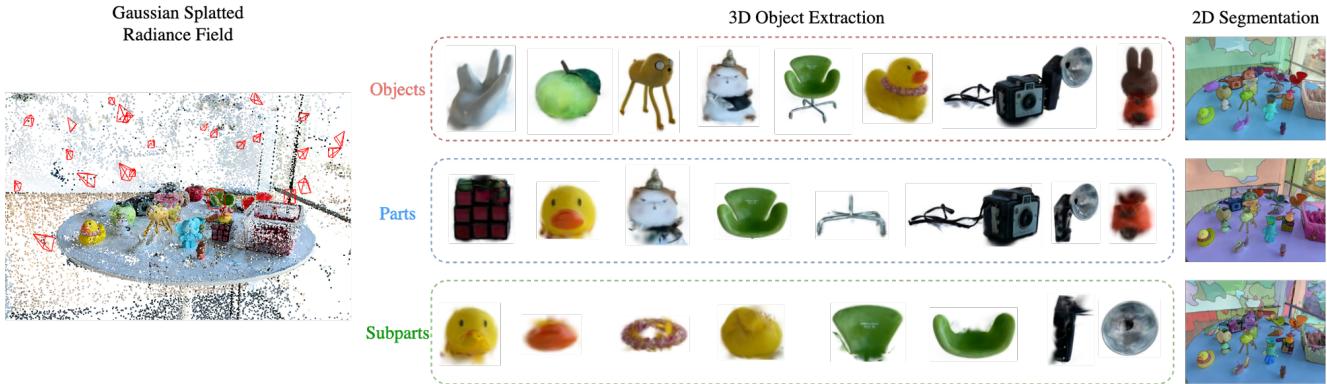


Figure 1. Lifting by Gaussians (LBG). LBG utilizes 2D foundation model masks to segment any pretrained 3DGS field into objects, parts, and subparts without gradient-based learning. For each frame, 2D segmentations are lifted onto the per-pixel max-contributor Gaussian, producing object fragments. These fragments are then merged into coherent, scene-level objects based on both geometric and semantic overlap. Through a hierarchical application of this process, LBG extracts high-quality 3D objects, parts, and subparts. In contrast to learning-based methods, LBG achieves this segmentation an order of magnitude faster, enabling new applications like object manipulation in augmented reality.

Abstract

We introduce *Lifting By Gaussians (LBG)*, a novel approach for open-world instance segmentation of 3D Gaussian Splatted Radiance Fields (3DGS). Recently, 3DGS Fields have emerged as a highly efficient and explicit alternative to Neural Field-based methods for high-quality Novel View Synthesis. Our 3D instance segmentation method leverages existing 2D foundation models like SAM, CLIP, and DINO to directly fuse 2D segmentation masks and dense features onto a 3DGS field. Unlike previous approaches, LBG requires no per-scene training, allowing it to operate seamlessly on any existing 3DGS reconstruction. Our approach is not only an order of magnitude faster and simpler than existing approaches; it is also highly modular, enabling 3D semantic segmentation of existing 3DGS fields without requiring a specific parametrization of the 3D Gaussians. Furthermore, our technique achieves superior semantic segmentation for 2D semantic novel view synthesis and 3D asset extraction results while maintaining

flexibility and efficiency. We further introduce a novel approach to evaluate individually segmented 3D assets from 3D radiance field segmentation methods.

1. Introduction

Semantic scene understanding — segmenting a 3D scene into its constituent objects — is a fundamental challenge in Computer Vision, with wide-ranging applications in Augmented and Virtual Reality (AR/VR), robotics, and autonomous vehicles. In this paper, we introduce a novel 3D scene segmentation method that leverages 2D semantic maps to segment any given 3DGS field [17].

2D segmentation has seen rapid advancements, particularly by developing robust 2D foundation models such as the Segment Anything model (SAM) [20]. However, creating a 3D foundation model that can similarly segment any 3D scene robustly has proven elusive due to the scarcity of annotated 3D data. To bypass the need for annotated

3D data, prior works on 3D segmentation [18, 36, 38, 51] instead opt to lift multi-view 2D image segmentation data onto 3D Neural Radiance Fields (NeRF) [32] or 3DGS [17]. Early techniques focused on a closed set of labels [7], while more recent work has leveraged open-vocabulary models like CLIP [37] and DINO features [1] for 2D segmentation. These approaches demonstrate that dense semantic labels optimized via inverse rendering-based formulations can effectively mitigate noisy ground-truth labels. While prior work has achieved great success in embedding 2D semantics onto 3D radiance fields, they either rely on expensive preprocessing steps to enforce multi-view consistency of the semantic images [46] or suffer from poor quality and long training times due to entanglement of 3D reconstruction and semantic segmentation [9, 19, 39, 47].

With an ever-growing corpus of existing 3DGS reconstructions, we are interested in quickly segmenting any existing Gaussian Radiance Field into its object, part, and subpart components. Our proposed method LBG accepts two inputs: 1) posed 2D image data, 2) a pre-trained 3DGS field. Using a 2D foundation model, we extract per-image 2D segmentation masks. We then employ a 2D-to-3D lifting approach to assign unique object IDs to Gaussians, creating per-image object fragments in 3D. We then use an incremental merging approach to sequentially merge these object fragments into coherent, scene-level objects. This method enables the segmentation of any existing 3DGS field with significantly reduced processing time compared to contrastive learning methods while delivering higher-quality results as seen in Table 1.

We validate our approach using standard benchmarks that assess the model’s ability to render high-quality semantic maps from novel views. However, similar to [21], we argue that the ultimate goal of 3D radiance field segmentation is to generate high-quality 3D assets rather than merely produce compelling 2D segmentation masks. Consequently, we introduce a new evaluation protocol that assesses the rendering quality of individually segmented 3D objects, providing a more accurate measure of 3D segmentation quality. Upon acceptance, we will release the code for our method and evaluation protocol along with our proposed 3D ground-truth datasets.

Our contributions can be summarized as follows:

- A training-free 3D instance segmentation approach that utilizes a novel 2D-to-3D lifting strategy to assign Gaussian semantics and an incremental merging procedure based on geometric and semantic overlap criteria. Our approach enables fast 3D segmentation of any existing 3DGS field without needing costly optimization of the 3D semantic field.
- A 3D segmentation refinement step that allows extraction of visually appealing 3D assets from existing

Table 1. Training time breakdown (in seconds) of state-of-the-art methods. All timings were benchmarked on a single RTX 3090 GPU. Preprocessing time includes loading the models, extracting 2D masks and computing foundation model features. Our method requires **10x** less time to achieve similar or higher 3D segmentation quality.

Methods	Preprocessing	3D Segmentation	Total
Gaussian Grouping [46]	293.44 s	3629.23 s	3922.67 s
SAGA [2]	1917.66 s	3289.08 s	5206.74 s
Ours	422.89 s	27.07 s	449.96 s

3DGS fields.

- A novel 3D semantic segmentation dataset and a quantitative evaluation protocol for evaluating the fidelity of individually extracted 3D assets.

2. Related Work

This section reviews key literature on 3D Gaussian Splatting for novel view synthesis and 3D scene segmentation techniques. For a comprehensive survey, we refer readers to [33].

Radiance Fields. Gaussian Splatting (3DGS) [17], introduced in 2023, has rapidly gained prominence as a real-time method for novel view synthesis, offering superior quality and speed compared to traditional Neural Radiance Fields (NeRFs) [32]. Since its inception, 3DGS has spurred research across various domains, including Simultaneous Localization and Mapping (SLAM) [13, 16, 30, 44], dynamic scene reconstruction [28, 43, 45], generative 3D/4D content creation [24, 27, 41], and meshing [10, 11, 48] among others. This surge of research has led to the development of various parameterizations of 3DGS [11, 17]. In this work, we introduce versatile semantic segmentation capabilities that can be seamlessly applied to any existing 3DGS field, regardless of the chosen parameterization, enabling semantic scene understanding for the growing corpus of 3DGS fields.

3D Scene Understanding. Understanding 3D scenes involves inferring the semantic properties of all objects within a scene — a fundamental challenge in 3D computer vision. Early approaches relied heavily on limited 3D ground truth data for tasks like object detection, localization, and segmentation [3, 4, 25]. To circumvent the scarcity of annotated 3D datasets, more recent work has increasingly leveraged 2D supervision [7], which gets fused onto the 3D representation. The advent of large foundation models [20, 37] has further enabled a shift from closed-set segmentation to an open-world framework, where a broader range of labels can be recognized and utilized [22, 23]. These developments in open-world segmentation of neural fields can be broadly categorized into feature distillation and mask-lifting techniques.

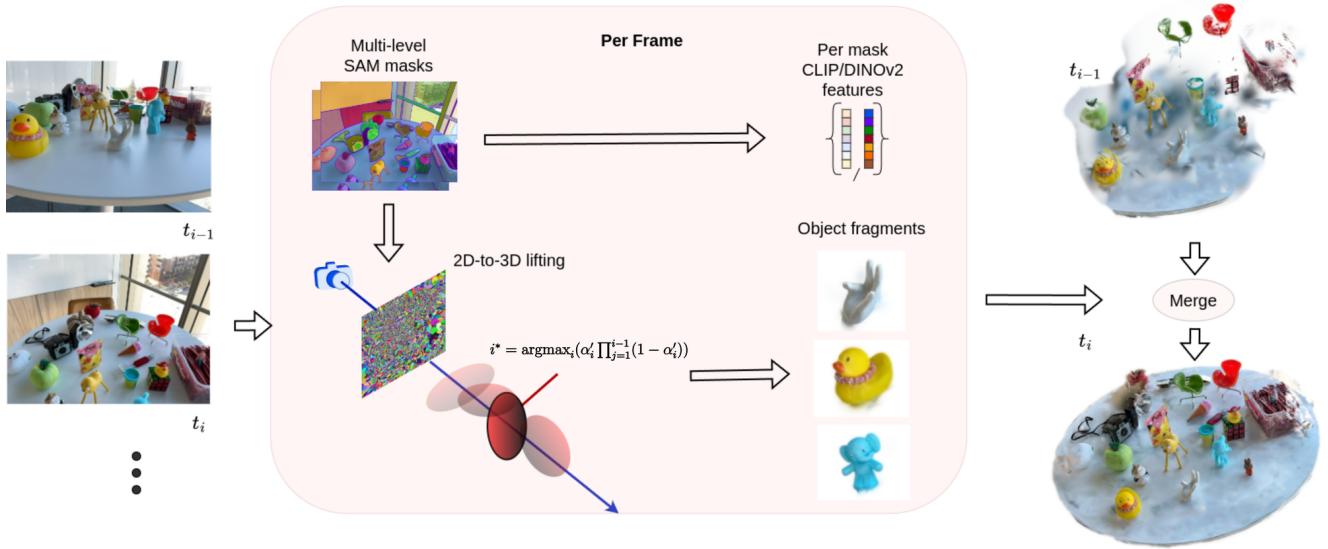


Figure 2. **LBG** constructs an open-vocabulary 3D instance segmentation from a sequence of posed RGB images. A generic 2D instance segmentation model is used to segment *objects*, *parts*, and *subparts* in each RGB image. Semantic feature vectors are extracted for each region, and the masks are lifted to the per-pixel max-contributing Gaussian, generating per-frame 3D object fragments. These fragments are incrementally merged into coherent, scene-level 3D objects. By applying this process hierarchically to the part and subpart masks, LBG produces a hierarchical decomposition of any 3DGS scene.

3D Segmentation through Feature distillation. Feature distillation [38, 51] aims to lift 2D features from computer vision foundation models such as CLIP [37], DINO [1]/DINO v2 [34] or SAM [20] onto the 3D representation. This is typically achieved through inverse rendering [38, 51] or feature aggregation [8, 14, 35, 40]. However, using inverse rendering to learn high-dimensional vectors on each Gaussians in a 3DGS field is computationally expensive, requiring substantial GPU memory and disk storage space, and often results in slower rendering speeds. Langsplat [36] mitigates some of these issues by compressing latent vectors specific to each scene, albeit at the cost of a lengthy preprocessing step. Moreover, direct aggregation on point clouds lacks the fidelity needed for photorealistic view synthesis. In contrast, our work demonstrates that high-quality 3D segmentation of 3DGS fields can be achieved without learning, offering a faster and more efficient alternative by directly aggregating semantics onto Gaussians.

3D Segmentation through 2D Mask lifting. Another line of research involves lifting 2D segmentation masks from foundation models like SAM [20] into 3D space [2, 9, 19, 39, 46, 47]. A primary challenge in these methods is achieving multi-view consistent segmentation masks. Gaussian Grouping [46] addresses this issue by using a tracking algorithm, though it struggles with errors under significant viewpoint changes. Other methods rely on contrastive learning to decompose and reconstruct the 3DGS

field with semantic information [9, 19, 39, 47]. These approaches are data-intensive, computationally expensive, and often entangle the 3D reconstruction with semantic learning. Our approach offers a more efficient alternative by fusing semantics directly onto the Gaussian field, significantly reducing computational costs and memory usage while maintaining or exceeding the segmentation quality of existing methods.

3. Method

We present *Lifting-by-Gaussians* (LBG), a novel and efficient method for rapidly lifting 2D semantic information onto any existing 3D Gaussian Splatting reconstruction. Given a set of posed RGB frames, our approach semantically decomposes a 3D environment within minutes. Using a 2D segmentation model, we identify object candidates and associate them across multiple views by applying semantic and geometric similarity measures to the lifted 3D Gaussians. To achieve fine-grained segmentation, we apply this lifting process hierarchically across multiple 2D segmentation scales. Our approach is illustrated in Figure 2.

A key advantage of LBG is its independence from gradient-based optimization, allowing seamless integration with any scene representation that uses 3D Gaussians, regardless of parameterization (e.g., 3DGS, 2DGS) and without needing to modify the underlying source code. Furthermore, our innovative 2D-to-3D lifting strategy enables di-

rect incorporation of pretrained 2D features, such as those from DINOv2 [34], CLIP [37], and SAM [20], onto the 3D Gaussians. This bypasses the need for costly training typically required to learn 3D-consistent features, making LBG a powerful and flexible solution for 3D scene understanding.

3.1. 3D Gaussian Splatting

3D Gaussian Splatting (3DGS) [17] represents a scene as a set of colored 3D Gaussian primitives. Unlike Neural Radiance Fields (NeRFs) [32], which have an implicit nature, 3DGS is an explicit representation, where each Gaussian g_i is parameterized by a position $\mu \in \mathbb{R}^3$, scale $S \in \mathbb{R}^3$, rotation $R \in \mathbb{R}^4$, opacity $\alpha \in [0, 1]$ and color $\mathbf{c} \in \mathbb{R}^3$ represented as three degrees of spherical harmonics (SH) coefficients. Images are rendered efficiently by splatting these 3D Gaussians onto the image plane using the approach from [52], and the resulting 2D projections are alpha-composited in a depth-first order. The color of a rendered pixel is then computed as:

$$c = \sum_{i=1}^N c_i \alpha'_i \prod_{j=1}^{i-1} (1 - \alpha'_j), \quad (1)$$

where $\alpha'_i = \alpha_i \cdot e^{-\frac{1}{2}(x' - \mu')^T \Sigma'^{-1} (x' - \mu')}$ defines the contribution of each splatted Gaussian to the pixel. The covariance matrix Σ is approximated as $\Sigma = RSS^T R^T$ to ensure positive semi-definiteness during optimization.

3.2. 2D-to-3D Lifting

Given a sequence of RGB images $\mathcal{I} = \{I^1, I^2, \dots, I^t\}$ and a pretrained 3DGS field \mathcal{G} , our goal is to generate a 3D semantic segmentation of the 3DGS field. LBG achieves this by incrementally creating a 3D semantic map of the scene. For each frame I^t at time, the Gaussian field \mathcal{G} is segmented into a set of objects \mathcal{O}^t . Each object $o_j^t = \langle \mathcal{G}_j^t, f_j^t \rangle$ is characterized by a set of Gaussians $\mathcal{G}_j^t \subset \mathcal{G}$ and a semantic feature vector f_j^t . Objects o_j^t from every new frame I^t are merged into the existing semantic map \mathcal{O}^{t-1} by either adding to existing objects or instantiating new ones.

2D Mask and Feature Extraction: LBG begins by extracting class-agnostic 2D segmentation masks $\{m_j^t\}_{j=1,\dots,M}$ using the Segment Anything (SAM) [20] model. SAM provides 2D segmentation masks at three semantic levels - *whole*, *part*, and *subpart*. For each extracted mask m_j^t we also extract semantic features f_j^t using CLIP [37] and DINO [34]. Since SAM masks lack inter-frame consistency, a mask fusion strategy is implemented using the learned 3D Gaussians to achieve consistent results across multiple frames.

Single frame 2D Mask to Gaussian Assignment:

Previous methods [29] have used pixel-Gaussian associations to lift 2D segmentations into 3D. However, these approaches typically assign object IDs to multiple alpha-blended Gaussians based on a threshold, leading to semantic bleeding artifacts. In contrast, LBG follows a pixel-to-Gaussian mapping inspired by [5], where each pixel is associated with the Gaussian that has the maximum alpha-blending weight. For each pixel $p \in m_j^t$, we identify the Gaussian $i^* = \arg \max_i (\alpha'_i \prod_{j=1}^{i-1} (1 - \alpha'_j))$ and assign a unique ID to both the 2D mask and its corresponding 3D Gaussians. This approach creates precise object segments o_j^t in 3D while minimizing semantic bleed-through.

Incremental Merging of 3D Object Fragments: For each new frame I^t , LBG merges the detected object fragments $o_j^t = \langle \mathcal{G}_j^t, f_j^t \rangle$ with the existing object map \mathcal{O}_j^{t-1} , constructed from the previous frames. We first compute the geometric overlap ratio between Gaussians belonging to object fragments from the current and previous frames as $\phi_{geom}(i, j) = \frac{|\mathcal{G}_i^t \cap \mathcal{G}_j^{t-1}|}{|\mathcal{G}_i^t|}$. Specifically, we count the number of overlapping Gaussians in both object maps. We further compute semantic similarity as the normalized cosine similarity between the feature vectors $\phi_{sem}(f_i^t, f_j^{t-1}) = \frac{f_i^t \cdot f_j^{t-1}}{2}$. Using these metrics, LBG greedily merges new object fragments with existing objects based on the highest similarity scores. If no suitable match is found, a new object is instantiated. Once merged, the Gaussian object segment ids are updated as $\mathcal{G}_i^t \cup \mathcal{G}_j^{t-1}$, and the semantic feature is updated using a running average: $f_{o_j} = \frac{n_{o_j} f_j^{t-1} + f_j^t}{n_{o_j} + 1}$, where n_{o_j} represents the number of fragments associated with object o_j so far.

Hierarchical Decomposition: Initially, LBG uses SAM’s *object-level* masks to create a high-level scene decomposition. Once the object map \mathcal{O}_t is constructed, each object is further split into parts and subparts using incremental merges applied at lower segmentation scales. This hierarchical decomposition is repeated across different levels of granularity, resulting in a scene graph that includes objects, parts, and subparts.

3.3. Post-Processing

While our maximum-contributor assignment strategy significantly reduces semantic bleeding, some label noise may persist. To generate clean 3D object assets, we include an optional post-processing step. This involves statistical outlier removal, similar to [2, 15, 46], alongside a split-

and-merge operation. Under-segmented fragments are split using 3D connected component analysis to identify salient clusters. Unassigned clusters are then merged with salient clusters based on their nearest-neighbor distance and overlap ratio, resulting in refined object segmentations.

4. Experiments

This section discusses the experimental setup, including datasets, evaluation metrics, and baselines. We then introduce our novel 3D asset segmentation evaluation protocol and show results on 2D mask rendering. Finally, we analyze our method’s design choices to justify our approach.

4.1. Experimental Setup

Datasets: We evaluate our method using two distinct datasets. The first is the LERF dataset [18], which features in-the-wild scenarios captured with a standard iPhone camera. The second dataset is 3D-OVS [26], which includes a collection of long-tail object categories. For evaluating 2D mask segmentation performance on the 3D-OVS dataset, we adhere to the evaluation protocol outlined in [26]. For the LERF dataset, we address labeling biases present in earlier annotations [36, 46], which previously focused on a limited number of central objects per scene. We re-annotate three scenes—*figurines*, *ramen*, and *teatime*—with a denser set of 2D instance labels. We annotate salient objects with 2D instance-level masks and open-world vocabulary annotations for our 2D mask rendering evaluations. Additionally, we use the LERF scenes in our 3D asset segmentation experiments. For these experiments, we first train a high-fidelity 3DGS field for each scene using [5] and then manually clean and refine the 3DGS field by selecting or removing Gaussians to generate a high-quality radiance field for each object.

Metrics: We assess our method’s performance on both 2D and 3D semantic segmentation tasks. For novel view synthesis of 2D masks, we use the established evaluation protocols from prior work [18, 29, 36, 46] and report the mean Intersection over Union (mIoU). In our 3D segmentation evaluations, we aim to measure how photorealistically a 3D segmented asset is compared to the underlying 3D model of the object. To evaluate this, we propose to 3D segment the 3DGS scene into individual objects and render 50 images of each object from various angles on a viewing hemisphere on a white background. To assess visual quality we compute perceptual similarity metrics such as peak signal-to-noise ratio (PSNR), structural similarity index (SSIM) [42], and Learned Perceptual Image Patch Similarity (LPIPS) [49] to compare the ground truth and predicted images.

Implementation Details: Our approach builds upon the Gaussian Splatting (3DGS) framework [17]. Through our experiments, we identified two key characteristics neces-

sary for accurate 3D asset extraction: minimal noise in the 3DGS field and fewer Gaussians to expedite object segmentation. To address these needs, we adopt Mini-Splatting [5] as our primary 3DGS representation. Mini-Splatting provides a compressed 3DGS model by significantly reducing the number of Gaussians through depth-based re-initialization and re-sampling, achieving a 10x reduction without degrading novel-view synthesis quality. We further enhance Mini-Splatting by incorporating a view consistency score to down-weight Gaussians visible from only a single camera, as these often include artifacts. Visualizations of our improved importance sampling can be found in the appendix.

For our experiments, we train Mini-Splatting [5] 3DGS for 30K iterations. Gaussian Grouping [46] is trained for 30K iterations across all scenes. For SAGA [2], we test two scenarios: 1) training a standard 3DGS representation for 30K iterations before training the semantic features of SAGA from scratch for 10K iterations and 2) training SAGA on top of our enhanced Mini-Splatting representation for 10K iterations.

Our 2D segmentation model uses SAM with the ViT-H backbone. For CLIP, we utilize the ViT-L/14 variant [37], and for DINOV2, we use the model described in [6].

Table 2. **Quanitative results on photorealistic 3D asset segmentation.** LBG outperforms previous approaches by better removing spurious Gaussians from the 3D object segmentation.

Dataset	Methods	PSNR ↑	SSIM ↑	LPIPS ↓
Figurines	Gaussian Grouping [46]	16.622	0.743	0.381
	SAGA [2]	19.574	0.836	0.243
	Ours	28.689	0.934	0.065
Ramen	Gaussian Grouping [46]	16.586	0.732	0.367
	SAGA [2]	15.570	0.693	0.444
	Ours	23.778	0.893	0.013
Teatime	Gaussian Grouping [46]	17.002	0.748	0.383
	SAGA [2]	16.072	0.740	0.395
	Ours	26.020	0.880	0.117

4.2. Photorealistic 3D Asset Extraction

Similar to [21], we argue that the ultimate goal of any 3D segmentation method is to extract clean, photorealistic 3D assets. However, the prevailing benchmark metrics in the literature mainly measure the ability of a method to render accurate 2D masks. We argue that rendering 2D masks, especially in the context of 3DGS, can hide small inaccuracies that average out through the alpha-blending process. We, therefore, propose a new evaluation protocol to measure how well any method performs in extracting photorealistic 3D assets. We use a subset of objects from the annotated 3DGS fields for this evaluation. We first match

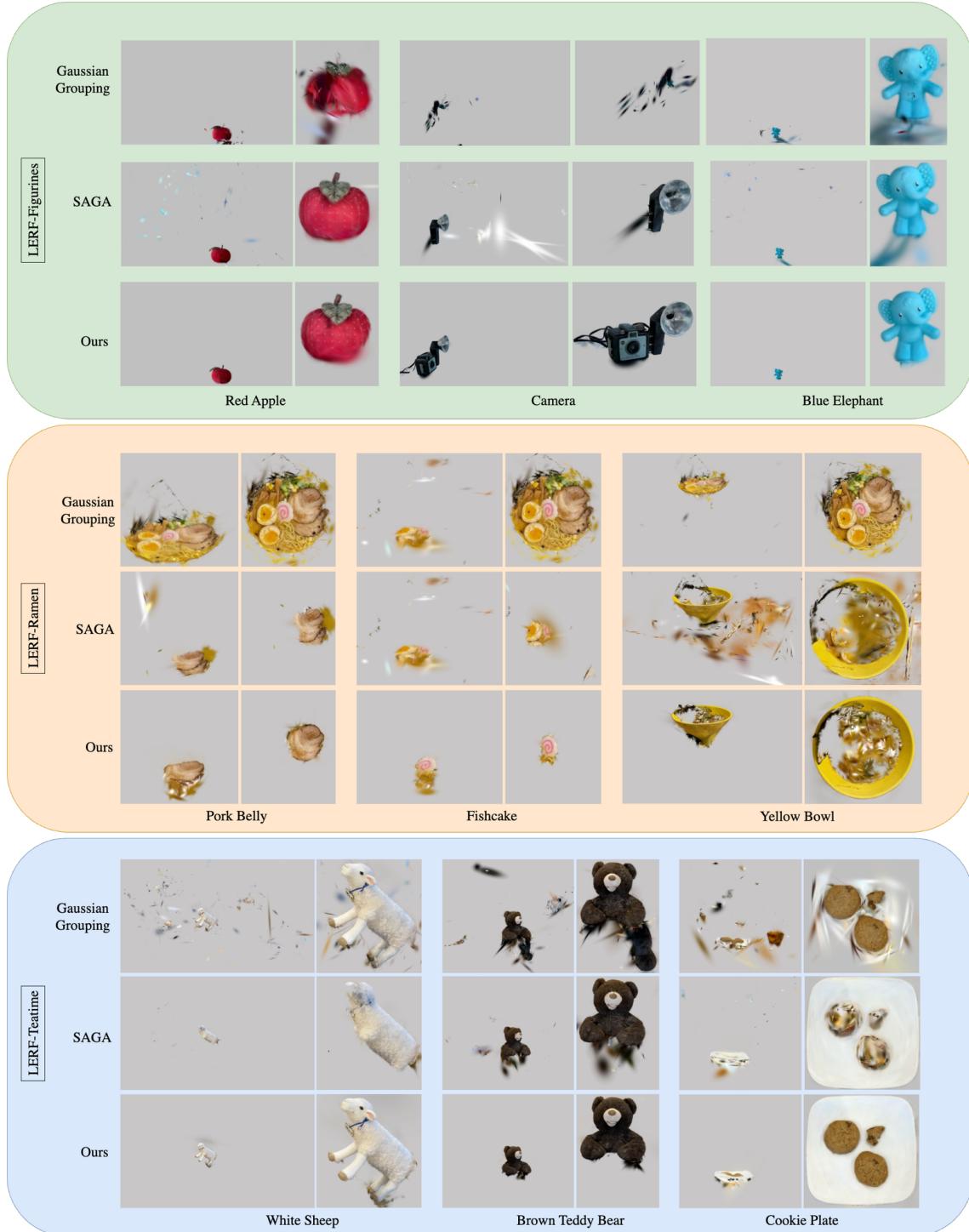


Figure 3. Qualitative comparison on the LERF dataset for 3D Asset extraction. We show three extracted objects per scene, with two different views for each object. Compared to prior methods, the objects extracted from LBG are much cleaner and have fewer noisy artifacts. 3D objects from SAGA and Gaussian Grouping have missing parts and are of lower quality overall.

ground-truth and predicted objects by comparing their rendered 2D mask projections across all training views and se-

lecting the one with the smallest average MSE. We extract 3D segmentations for Gaussian Grouping by applying the

Table 3. **2D Novel View Synthesis of 2D Instance Segmentation Masks.** We report mIoU (\uparrow) on the LERF dataset [18] (left) and the 3D-OVS dataset [26]. Best results are highlighted as **first**, **second** and **third**. In contrast to the baselines, our method was **not optimized** for this task, but it still performs comparably. Numbers with * are taken from [2]

Methods	<i>figurines</i>	<i>ramen</i>	<i>teatime</i>
Gaussian Grouping [46]	0.697	0.458	0.619
SAGA-MS [2]	0.838	0.583	0.673
SAGA-3DGS [2]	0.860	0.803	0.874
Ours	0.822	0.732	0.866

learned object classifier on each 3D Gaussian. For SAGA, we follow the proposed protocol [2] of clustering the 3D feature field using HDBSCAN [31].

As shown in Figure 3 and Table 2, our method outperforms all baselines on this task by a large margin. This can be largely attributed to our simple but effective lifting strategy that avoids bleeding artifacts arising from alpha-blending-based learning on 2D features. Additionally, our merge procedure relies on spatial and feature proximity. By only looking at objects accumulated until the previous frame for the merge, we ensure that disconnected Gaussians do not receive spurious labels. Gaussian Grouping suffers from noisy mask predictions and inconsistent segmentation scales. For example, in the *ramen* scene, the method fails to distinguish between objects in the ramen bowl. Similarly in the *figurines* and *teatime* scenes, Gaussian Grouping produces very noisy 3D assets that do not resemble coherent objects. In contrast, SAGA shows much cleaner boundaries in 3D. However, the absence of a well-defined scale hierarchy produces incomplete segmentations like the partial camera segmentation in *figurines*. Consequently, SAGA often tends to over-segment objects into much smaller clusters. Additionally, 3D segmentations extracted from SAGA contain many spurious Gaussians, as can be seen across all three scenes in Figure 3.

4.3. Novel View Synthesis of 2D Instance Segmentation Masks

We further evaluate LBG on novel view synthesis of 2D instance masks on two datasets, LERF [18] and 3D-OVS [26]. On the LERF dataset, we compare our method against Gaussian Grouping [46] and SAGA [2]. As each method produces 3D instance segmentation labels differently, we first match ground truth masks to predicted masks by finding the prediction with maximum IoU overlap. To compare fine-grained instance segmentation, we extract multi-level masks where possible. Gaussian Grouping does not provide a hierarchical decomposition of the scene; we only use object-level masks. For SAGA, we compute three segmentation levels, as shown in their paper, namely 0.1, 0.5, and 1.0. For our method, we use the object, part, and

Methods	Avg.	<i>bed</i>	<i>bench</i>	<i>room</i>	<i>lawn</i>	<i>sofa</i>
LSEG [22]	56.0	6.0	19.2	4.5	17.5	20.6
OVSeg [23]	79.8	88.9	71.4	66.1	81.2	77.5
LERF [18]	73.5	53.2	46.6	27.0	73.7	54.8
3D-OVS [26]	89.5	89.3	92.8	74.0	88.2	86.8
LangSplat [36]	73.02	77.8	77.3	58.4	90.9	60.2
Gaussian Grouping [46]	88.96	64.5	95.6	96.4	97.0	91.3
SAGA [2] *	96.0	97.4	95.4	96.8	96.6	93.5
Ours	94.9	97.7	96.3	95.9	97.3	87.4

subpart hierarchy.

In Table 3 and Figure 4 we show quantitative and qualitative results. Note that while all the baselines are optimized to perform well on the 2D instance mask novel view synthesis task, ours is not. Even so, our method shows competitive performance across the board compared to other approaches (Table 3). As in the 3D segmentation case, Gaussian Grouping can sometimes not distinguish between different object scales, such as in the Ramen bowl in Figure 4 (middle). Comparing SAGA and our method, we see that both generate different failure cases. SAGA fails to detect some small objects like the spatula and camera body in *figurines* and part of the pork belly and green onions in *ramen* entirely. Our method generates a complete segmentation map. However, in some cases, LBG fails to achieve the desired segmentation granularity. For example, in the *figurines* scene, our method segments the container, Twizzlers, and Waldo together. This can be attributed to inconsistent object segmentation masks from SAM. For additional qualitative results, please see the appendix.

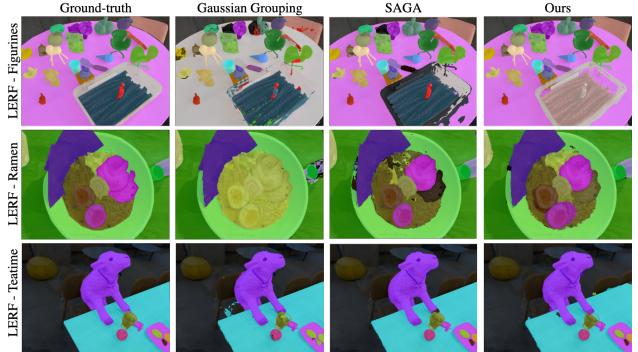


Figure 4. **Qualitative comparison on novel view synthesis for 2D instance masks.** Black regions are unassigned. We see that our 2D masks are on par with other methods. LBG picks out instances across segmentation scales better than Gaussian Grouping. Compared to SAGA, our method provides more complete masks.

4.4. Ablations

We conduct a series of ablations in Table 4 to evaluate the effect of each design component in LBG. Specifically, we ablate the features of the foundation model used to compute the similarity score in the merging step and the 3DGS reconstruction methods. From Figure 5, we see that feature selection is critical in merging correct objects. Our post-processing that filters outlier Gaussians provides a slight improvement in segmentation quality. Finally, we evaluate our method with a different choice of a segmentation model [50].

Table 4. **Ablation experiments.** We show the impact of different design choices on the final 2D instance segmentation result. We report mIoU (\uparrow) on the LERF *figurines* dataset.

Ablation	mIoU
Ours, full	0.822
w/o filtering	0.815
w/o CLIP feat.	0.782
w/o minisplatting GS	0.781
w/o DINO feat.	0.779
w/ Fast-SAM [50]	0.608

Effect of post-processing: Quantitatively, post-processing has only a minimal effect on the 2D instance segmentation results. However, we observe certain cases, where the post-processing step correctly cleans up over-segmented regions and merges them with the correct object.

Effect of CLIP and DINO features: Using just the exact Gaussian overlap as a similarity score tends to aggregate partially overlapping objects as shown in Fig. 5 disregarding their semantic difference. This issue is particularly pronounced for smaller objects, which tend to merge into larger object clusters nearby. This issue is alleviated when considering feature similarity as part of the similarity score.

Choice of 3DGS representation: We observe that the choice of 3DGS representation significantly impacts the final performance. This can be attributed to the post-processing step that scales with the number of Gaussians in the scene. Minisplatting reduces floaters and cloudy artifacts prevalent in vanilla 3DGS reconstructions and has 10x fewer Gaussians. This, in turn, reduces the number of mislabeled Gaussians. Additionally, a smaller GS field allows our algorithm to run much quicker.

Choice of segmentation model: Since our approach is agnostic to the segmentation model used, we apply a faster SAM-variant in our mask extraction step. This speeds

up our performance by 4x while providing reasonable 3D segmentations. For more results, please see the appendix.

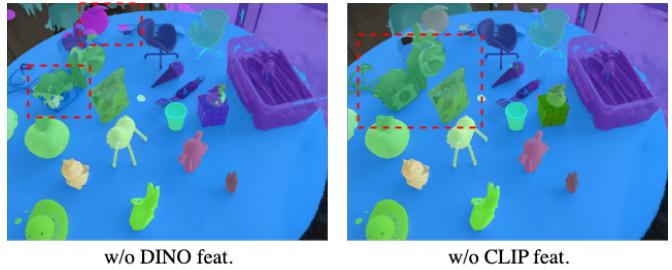


Figure 5. **Ablation on using CLIP features for merging.** Using only spatial proximity leads to nearby objects being grouped together (red dashed boxes). When using DINO features together with CLIP this error is fixed.

5. Conclusion

In this work, we presented LBG, a novel framework for generating high-quality 3D segmentations from any trained 3D Gaussian Splatting (3DGS) field. Our approach uniquely lifts off-the-shelf 2D segmentation masks onto the corresponding per-pixel max-contributor Gaussians, followed by an incremental merging process that consolidates object fragments across frames. By leveraging the max-contributor Gaussians for this 2D-to-3D lifting, LBG significantly mitigates semantic bleeding issues that have plagued prior methods, ensuring more accurate and clean object boundaries in the 3D domain.

Experimental results demonstrate that LBG not only produces superior 3D assets but also does so at a speed that is an order of magnitude faster than state-of-the-art methods. This substantial improvement in efficiency, combined with high-quality outputs, highlights the transformative potential of LBG for large-scale 3D scene segmentation and reconstruction tasks. Moreover, despite not being specifically optimized for novel view synthesis of 2D instance masks, our method delivers competitive performance on 2D instance mask generation, showcasing its versatility across multiple applications.

5.1. Limitations and Future Work

While LBG achieves efficient, high-quality results, several limitations remain. First, the model loading times hinder real-time applications, which future work could address through optimization or model compression techniques. Second, our method occasionally struggles with segmenting small objects, as the current merging approach may not capture them. A potential solution would involve incorporating a fine-tuning step with minimal training iterations to refine these initial segmentations.

References

- [1] 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 International Conference on Computer Vision (ICCV)*, 2021. 2, 3
- [2] Jiazhong Cen, Jiemin Fang, Chen Yang, Lingxi Xie, Xiaopeng Zhang, Wei Shen, and Qi Tian. Segment Any 3D Gaussians. *arXiv preprint arXiv:2312.00860*, 2023. 2, 3, 4, 5, 7, 12
- [3] Dave Zhenyu Chen, Angel X Chang, and Matthias Nießner. Scanrefer: 3d object localization in rgb-d scans using natural language. In *European conference on computer vision*, pages 202–221. Springer, 2020. 2
- [4] Angela Dai, Angel X. Chang, Manolis Savva, Maciej Halber, Thomas A. Funkhouser, and Matthias Nießner. ScanNet: Richly-Annotated 3D Reconstructions of Indoor Scenes. *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 2432–2443, 2017. 2
- [5] Guangchi Fang and Bing Wang. Mini-Splatting: Representing Scenes with a Constrained Number of Gaussians, 2024. *eprint: 2403.14166*. 4, 5
- [6] Stephanie Fu, Mark Hamilton, Laura E. Brandt, Axel Feldmann, Zhoutong Zhang, and William T. Freeman. FeatUp: A Model-Agnostic Framework for Features at Any Resolution. In *The Twelfth International Conference on Learning Representations*, 2024. 5
- [7] Kyle Genova, Xiaoqi Yin, Abhijit Kundu, Caroline Pantofaru, Forrester Cole, Avneesh Sud, Brian Brewington, Brian Shucker, and Thomas Funkhouser. Learning 3d semantic segmentation with only 2d image supervision. In *2021 International Conference on 3D Vision (3DV)*, pages 361–372. IEEE, 2021. 2
- [8] Qiao Gu, Ali Kuwajerwala, Sacha Morin, Krishna Murthy Jatavallabhula, Bipasha Sen, Aditya Agarwal, Corban Rivera, William Paul, Kirsty Ellis, Rama Chellappa, and others. Conceptgraphs: Open-vocabulary 3d scene graphs for perception and planning. In *2024 IEEE International Conference on Robotics and Automation (ICRA)*, pages 5021–5028. IEEE, 2024. 3
- [9] Qiao Gu, Zhaoyang Lv, Duncan Frost, Simon Green, Julian Straub, and Chris Sweeney. EgoLifter: Open-world 3D Segmentation for Egocentric Perception. *arXiv preprint arXiv:2403.18118*, 2024. 2, 3
- [10] Antoine Guédon and Vincent Lepetit. Sugar: Surface-aligned gaussian splatting for efficient 3d mesh reconstruction and high-quality mesh rendering. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 5354–5363, 2024. 2
- [11] Binbin Huang, Zehao Yu, Anpei Chen, Andreas Geiger, and Shenghua Gao. 2d gaussian splatting for geometrically accurate radiance fields. In *ACM SIGGRAPH 2024 Conference Papers*, pages 1–11, 2024. 2
- [12] Binbin Huang, Zehao Yu, Anpei Chen, Andreas Geiger, and Shenghua Gao. 2d gaussian splatting for geometrically accurate radiance fields. In *SIGGRAPH 2024 Conference Papers*, 2024. 14
- [13] Huajian Huang, Longwei Li, Hui Cheng, and Sai-Kit Yeung. Photo-SLAM: Real-time Simultaneous Localization and Photorealistic Mapping for Monocular Stereo and RGB-D Cameras. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 21584–21593, 2024. 2
- [14] Krishna Murthy Jatavallabhula, Alihusein Kuwajerwala, Qiao Gu, Mohd Obama, Tao Chen, Shuang Li, Ganesh Iyer, Soroush Saryazdi, Nikhil Keetha, Ayush Tewari, Joshua B. Tenenbaum, Celso Miguel de Melo, Madhava Krishna, Liam Paull, Florian Shkurti, and Antonio Torralba. ConceptFusion: Open-set Multimodal 3D Mapping. *Robotics: Science and Systems (RSS)*, 2023. 3
- [15] Shengxiang Ji, Guanjun Wu, Jiemin Fang, Jiazhong Cen, Taoran Yi, Wenyu Liu, Qi Tian, and Xinggang Wang. Segment Any 4D Gaussians, 2024. *eprint: 2407.04504*. 4
- [16] Nikhil Keetha, Jay Karhade, Krishna Murthy Jatavallabhula, Gengshan Yang, Sebastian Scherer, Deva Ramanan, and Jonathon Luiten. SplatTAM: Splat Track & Map 3D Gaussians for Dense RGB-D SLAM. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 21357–21366, 2024. 2
- [17] Bernhard Kerbl, Georgios Kopanas, Thomas Leimkuhler, and George Drettakis. 3D Gaussian Splatting for Real-Time Radiance Field Rendering. *ACM Transactions on Graphics (TOG)*, 42(4):1–14, 2023. Publisher: ACM New York, NY, USA. 1, 2, 4, 5
- [18] Justin Kerr, Chung Min Kim, Ken Goldberg, Angjoo Kanazawa, and Matthew Tancik. Lerf: Language embedded radiance fields. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 19729–19739, 2023. 2, 5, 7
- [19] Chung Min Kim, Mingxuan Wu, Justin Kerr, Ken Goldberg, Matthew Tancik, and Angjoo Kanazawa. Garfield: Group anything with radiance fields. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 21530–21539, 2024. 2, 3
- [20] Alexander Kirillov, Eric Mintun, Nikhila Ravi, Hanzi Mao, Chloe Rolland, Laura Gustafson, Tete Xiao, Spencer Whitehead, Alexander C Berg, Wan-Yen Lo, and others. Segment anything. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 4015–4026, 2023. 1, 2, 3, 4
- [21] Hyunjee Lee, Youngsik Yun, Jeongmin Bae, Seoha Kim, and Youngjung Uh. Rethinking Open-Vocabulary Segmentation of Radiance Fields in 3D Space. *arXiv preprint arXiv:2408.07416*, 2024. 2, 5
- [22] Boyi Li, Kilian Q. Weinberger, Serge Belongie, Vladlen Koltun, and Rene Ranftl. Language-driven Semantic Segmentation. In *International Conference on Learning Representations*, 2022. 2, 7, 14
- [23] Feng Liang, Bichen Wu, Xiaoliang Dai, Kunpeng Li, Yinan Zhao, Hang Zhang, Peizhao Zhang, Peter Vajda, and Diana Marculescu. Open-vocabulary semantic segmentation with mask-adapted clip. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 7061–7070, 2023. 2, 7, 14

- [24] Yixun Liang, Xin Yang, Jiantao Lin, Haodong Li, Xiaogang Xu, and Yingcong Chen. Luciddreamer: Towards high-fidelity text-to-3d generation via interval score matching. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 6517–6526, 2024. 2
- [25] Yiyi Liao, Jun Xie, and Andreas Geiger. Kitti-360: A novel dataset and benchmarks for urban scene understanding in 2d and 3d. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 45(3):3292–3310, 2022. Publisher: IEEE. 2
- [26] Kunhao Liu, Fangneng Zhan, Jiahui Zhang, Muyu Xu, Yingchen Yu, Abdulmotaleb El Saddik, Christian Theobalt, Eric Xing, and Shijian Lu. Weakly supervised 3d open-vocabulary segmentation. *Advances in Neural Information Processing Systems*, 36:53433–53456, 2023. 5, 7, 12
- [27] Xian Liu, Xiaohang Zhan, Jiaxiang Tang, Ying Shan, Gang Zeng, Dahua Lin, Xihui Liu, and Ziwei Liu. Humangaussian: Text-driven 3d human generation with gaussian splatting. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 6646–6657, 2024. 2
- [28] Jonathon Luiten, Georgios Kopanas, Bastian Leibe, and Deva Ramanan. Dynamic 3d gaussians: Tracking by persistent dynamic view synthesis. In *2024 International Conference on 3D Vision (3DV)*, pages 800–809. IEEE, 2024. 2
- [29] Weijie Lyu, Xuetong Li, Abhijit Kundu, Yi-Hsuan Tsai, and Ming-Hsuan Yang. Gaga: Group Any Gaussians via 3D-aware Memory Bank, 2024. eprint: 2404.07977. 4, 5
- [30] Hidenobu Matsuki, Riku Murai, Paul HJ Kelly, and Andrew J Davison. Gaussian splatting slam. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 18039–18048, 2024. 2
- [31] Leland McInnes, John Healy, Steve Astels, and others. hdb-scan: Hierarchical density based clustering. *J. Open Source Softw.*, 2(11):205, 2017. 7
- [32] 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. Publisher: ACM New York, NY, USA. 2, 4
- [33] Thang-Anh-Quan Nguyen, Amine Bourki, M’aty’as Macudzinski, Anthony Brunel, and Mohammed Bennamoun. Semantically-aware Neural Radiance Fields for Visual Scene Understanding: A Comprehensive Review. *ArXiv*, abs/2402.11141, 2024. 2
- [34] Maxime Oquab, Timothée Darcet, Theo Moutakanni, Huy V. Vo, Marc Szafraniec, Vasil Khalidov, Pierre Fernandez, Daniel Haziza, Francisco Massa, Alaaeldin El-Nouby, Russell Howes, Po-Yao Huang, Hu Xu, Vasu Sharma, Shang-Wen Li, Wojciech Galuba, Mike Rabbat, Mido Assran, Nicolas Ballas, Gabriel Synnaeve, Ishan Misra, Herve Jegou, Julien Mairal, Patrick Labatut, Armand Joulin, and Piotr Bojanowski. DINoV2: Learning Robust Visual Features without Supervision, 2023. Publication Title: arXiv:2304.07193. 3, 4, 15
- [35] Songyou Peng, Kyle Genova, Chiyu ”Max” Jiang, Andrea Tagliasacchi, Marc Pollefeys, and Thomas Funkhouser. OpenScene: 3D Scene Understanding with Open Vocabularies. In *CVPR*, 2023. 3
- [36] 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, 5, 7, 14
- [37] Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, and others. Learning transferable visual models from natural language supervision. In *International conference on machine learning*, pages 8748–8763. PMLR, 2021. 2, 3, 4, 5
- [38] William Shen, Ge Yang, Alan Yu, Jansen Wong, Leslie Pack Kaelbling, and Phillip Isola. Distilled Feature Fields Enable Few-Shot Language-Guided Manipulation. In *7th Annual Conference on Robot Learning*, 2023. 2, 3
- [39] Myrna C Silva, Mahtab Dahaghin, Matteo Toso, and Alessio Del Bue. Contrastive Gaussian Clustering: Weakly Supervised 3D Scene Segmentation. *arXiv preprint arXiv:2404.12784*, 2024. 2, 3
- [40] Ayça Takmaz, Elisabetta Fedele, Robert W. Sumner, Marc Pollefeys, Federico Tombari, and Francis Engelmann. OpenMask3D: Open-Vocabulary 3D Instance Segmentation. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2023. 3
- [41] Jiaxiang Tang, Jiawei Ren, Hang Zhou, Ziwei Liu, and Gang Zeng. DreamGaussian: Generative Gaussian Splatting for Efficient 3D Content Creation. *The Twelfth International Conference on Learning Representations*, 2024. 2
- [42] Zhou Wang, Alan C Bovik, Hamid R Sheikh, and Eero P Simoncelli. Image quality assessment: from error visibility to structural similarity. *IEEE transactions on image processing*, 13(4):600–612, 2004. 5
- [43] Guanjun Wu, Taoran Yi, Jiemin Fang, Lingxi Xie, Xiaopeng Zhang, Wei Wei, Wenyu Liu, Qi Tian, and Xinggang Wang. 4d gaussian splatting for real-time dynamic scene rendering. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 20310–20320, 2024. 2
- [44] 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
- [45] Ziyi Yang, Xinyu Gao, Wen Zhou, Shaohui Jiao, Yuqing Zhang, and Xiaogang Jin. Deformable 3d gaussians for high-fidelity monocular dynamic scene reconstruction. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 20331–20341, 2024. 2
- [46] Mingqiao Ye, Martin Danelljan, Fisher Yu, and Lei Ke. Gaussian grouping: Segment and edit anything in 3d scenes. *European Conference on Computer Vision (ECCV)*, 2023. 2, 3, 4, 5, 7
- [47] Haiyang Ying, Yixuan Yin, Jinzhi Zhang, Fan Wang, Tao Yu, Ruqi Huang, and Lu Fang. Omniseg3d: Omnipractical 3d segmentation via hierarchical contrastive learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 20612–20622, 2024. 2, 3

- [48] Baowen Zhang, Chuan Fang, Rakesh Shrestha, Yixun Liang, Xiaoxiao Long, and Ping Tan. RaDe-GS: Rasterizing Depth in Gaussian Splatting. *arXiv preprint arXiv:2406.01467*, 2024. 2
- [49] Richard Zhang, Phillip Isola, Alexei A Efros, Eli Shechtman, and Oliver Wang. The Unreasonable Effectiveness of Deep Features as a Perceptual Metric. In *CVPR*, 2018. 5
- [50] Xu Zhao, Wenchao Ding, Yongqi An, Yinglong Du, Tao Yu, Min Li, Ming Tang, and Jinqiao Wang. Fast Segment Anything, 2023. *eprint: 2306.12156*. 8
- [51] Shijie Zhou, Haoran Chang, Sicheng Jiang, Zhiwen Fan, Zehao Zhu, Dejia Xu, Pradyumna Chari, Suya You, Zhangyang Wang, and Achuta Kadambi. Feature 3dgs: Supercharging 3d gaussian splatting to enable distilled feature fields. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 21676–21685, 2024. 2, 3
- [52] Matthias Zwicker, Hanspeter Pfister, Jeroen Van Baar, and Markus Gross. EWA volume splatting. In *Proceedings Visualization, 2001. VIS'01.*, pages 29–538. IEEE, 2001. 4

A. Appendix

In this appendix, we provide further experimental results, including additional 3D segmentation comparisons in Section B, a qualitative comparison with SAGA [2] on rendering of 2D masks at novel views through different scales in Section C.1 and qualitative results on the 3D-OVS [26] dataset in Section C.2. We further show that our method can be used to 3D segment 2DGS fields without modification in Section D. Finally, we show some intuition into our improvements of the Mini-Splatting importance sampling in Section F and conclude by showing an application of our method to lift 2D feature maps, such as DINO, into 3D in Section G.

B. Additional 3D results

We show additional results of extracted 3D objects from our LBG method in Figure 6. Contrastive methods like SAGA require a 3D feature clustering step to extract objects, which is prone to floaters and noise. Gaussian Grouping also requires a 3D clustering step which produces noisy 3D objects. Our 3D objects are more coherent and have cleaner boundaries than other methods due to our simple lifting and mask merging strategy.

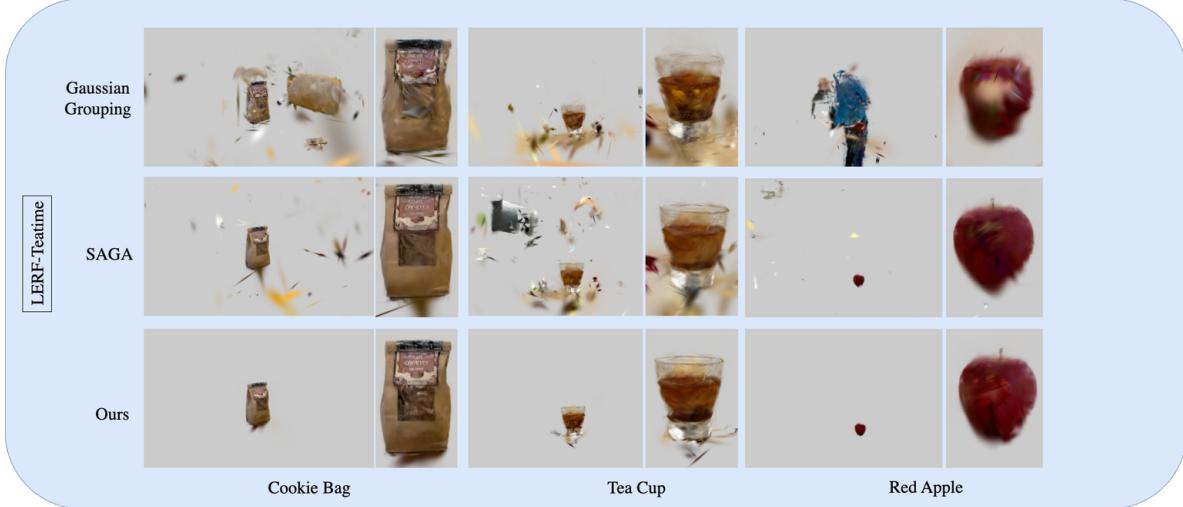
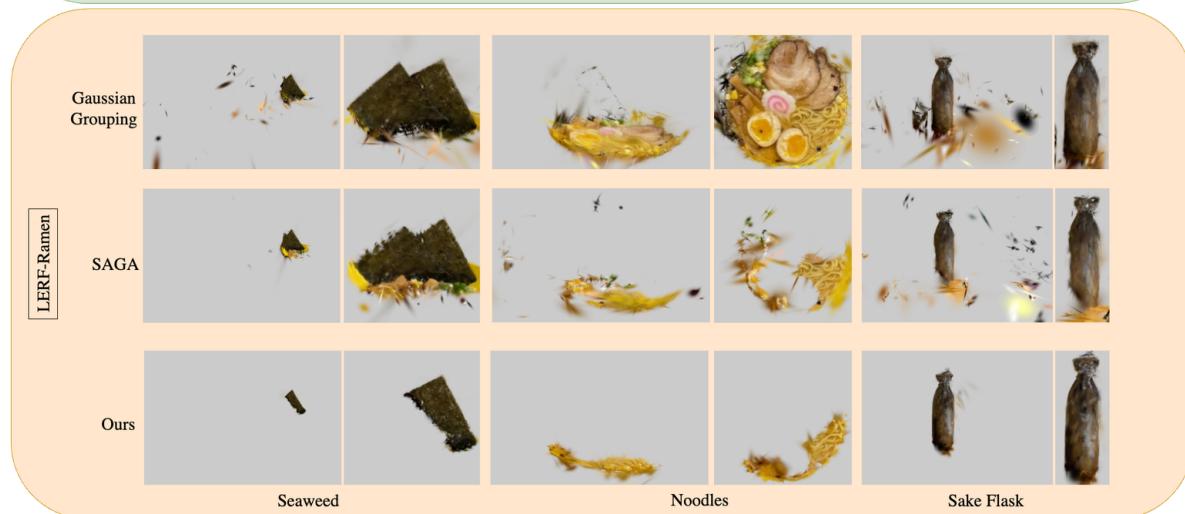
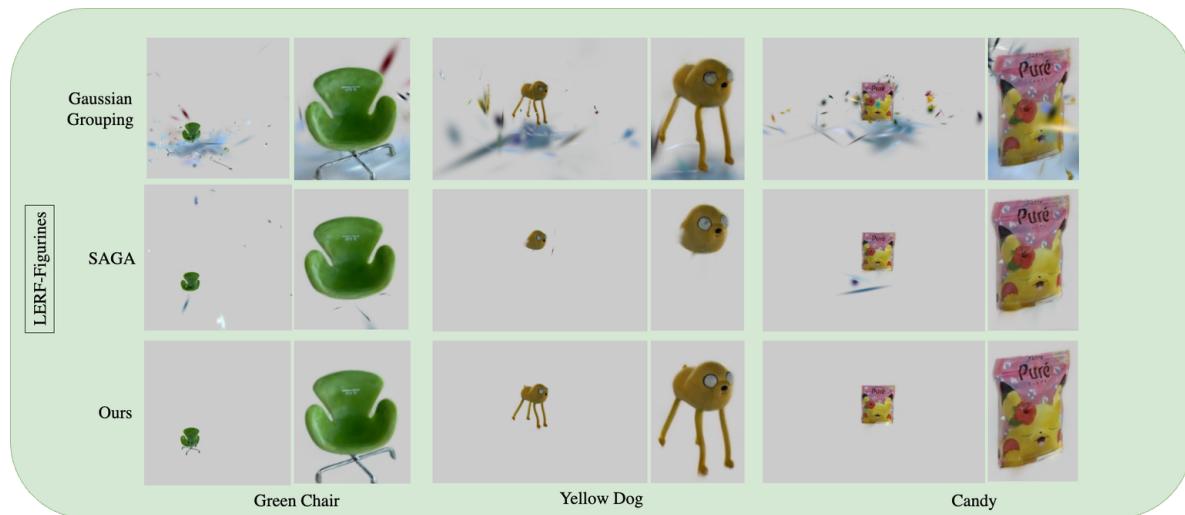


Figure 6. Additional 3D segmentation results on LERF dataset.

C. Additional 2D results

C.1. LERF

We show mask novel view synthesis results on the three LERF scenes in Figure 7. Specifically, we compare SAGA and LBG. For SAGA, we show images rendered at three levels: 0.1 (left), 0.5 (middle), and 1.0 (right). For our method, we show object level (left), part level (middle), and subpart level (right).

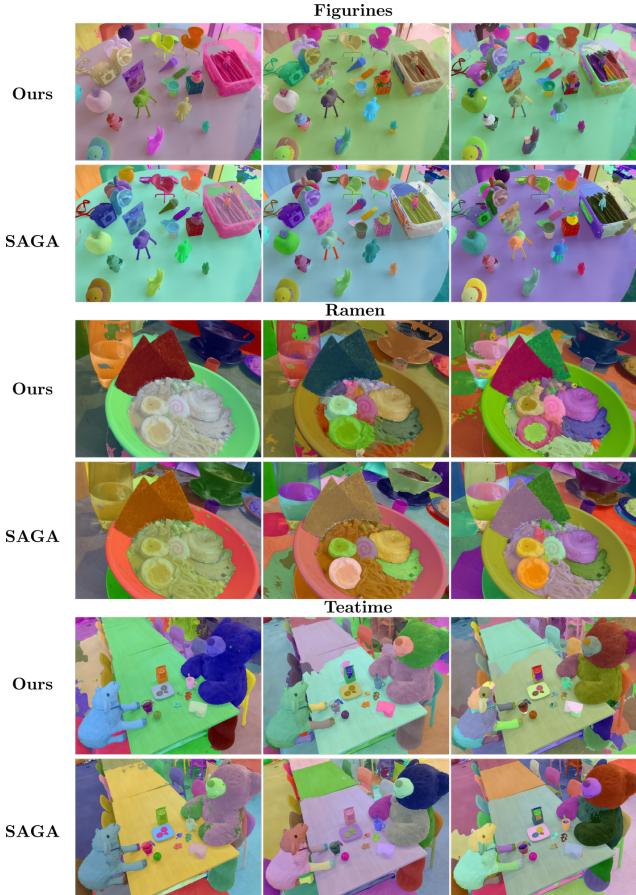


Figure 7. Additional results on novel view synthesis for 2D instance masks. For SAGA, we show images rendered at three levels: 0.1 (left), 0.5 (middle), and 1.0 (right). For our method, we show object level (left), part level (middle), and subpart level (right).

Even though our method is not optimized on the task of novel view synthesis for 2D masks, it performs well, especially on the object level. We can see that SAGA often breaks up objects into parts, even on the top level (camera in figurines, bear in teatime). This is largely due to SAGA using metric diagonal measurements to determine scale without associating these scales back to object/part/subpart decompositions. We argue that instead of using such arbitrary scales it is much more intuitive to break a scene into its

logical parts, starting from complete objects.

C.2. 3DOVS

We compare our method for segmentation against prior methods, such as LSEG [22] and OVSeg [23]. The numbers for these methods are taken from [36]. We found that LangSplat evaluations, as described in the paper, led to sub-optimal performance due to limited contrast in the learned feature representation. To improve performance, we modified the protocol described in the paper and used a per-scene threshold.

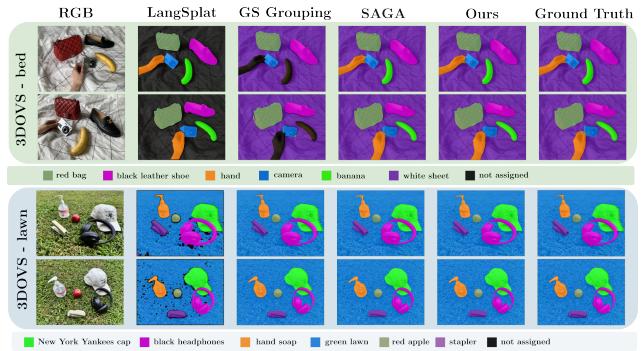


Figure 8. Qualitative comparison on the 3DOVS dataset. Black regions are unassigned. In the bed scene, Gaussian Grouping merges hand and banana objects together, resulting in segmentation failure. Similarly, LangSplat fails to segment the white sheet due to low contrast in the feature space. Our method shows cleaner boundaries compared to both baselines.

On the 3DOVS dataset (Fig. 8), our method demonstrates superior performance across the board against most methods and is comparable to SAGA. Notably, LangSplat overlooks the background in the bed scene and exhibits gaps in the lawn scene, attributed to inconsistencies in thresholds and noise within the *subpart* level of the language embeddings. While Gaussian Grouping yields results similar to our method, it often produces less defined boundaries due to tendencies towards over-segmentation. Regions in black are segmentations that were not detected during the evaluation.

D. Applying our method on 2DGS

As our method, LBG can consume any Gaussian Splatting-based reconstruction, we apply our method on a scene reconstructed using 2DGS [12] without any modification. As a consequence of using [12], we can produce meshes of the individual segmented objects. Results are shown in Figure 9.

E. Additional Ablations

We present additional ablation results using our method with Fast-SAM instead of the standard Segment Anything

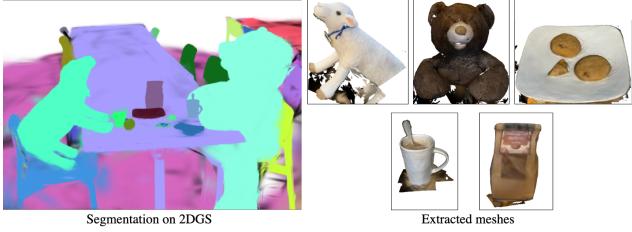


Figure 9. **LBG Segmentation on 2DGS.** 2DGS with colored Gaussians according to instance IDs (left) and individually extracted meshes (right).

Model for mask extraction. While the Fast-SAM model provides results in near real-time, which is desirable for most applications in robotics and AR, Figure 10 shows that the results are much worse. We can see that the segmentation lifted with Fast-SAM masks struggles to keep clean object boundaries. Furthermore, even with our post-processing step that merges adjoining clusters, we can see that the Fast-SAM model still contains many objects that cannot be merged using a purely geometric approach.



Figure 10. **Additional ablation results.** We show performance of our method using the standard SAM model (left) and Fast-SAM (right).

F. Improvements to Mini-Splatting

We adopted a technique similar to Mini-splatting to remove floaters from Gaussian Splatting reconstructions. 3D Gaussians are removed based on a probability dictated by an importance score. Initially, we found that using opacity contribution as the basis for this score was insufficient, as it assigned small values to floaters. Many floaters, we discovered, resulted from over-fitting to a single view (see Figure 11). To address this, we augmented the probability score by considering the number of views a 3D Gaussian maximally contributes to, through a log multiplier on the number of views. This modification, combined with the pruning and resampling strategy from Mini-splatting, effectively reduces floaters, particularly those caused by single-view over-fitting.

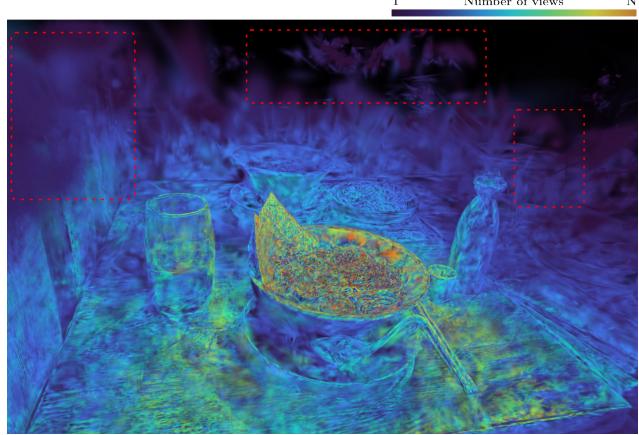


Figure 11. **Visualization of the number of views which see each Gaussian.** Notice how many structured floaters are only seen by a single view, showcasing visual artifacts from single-view overfitting.

G. DINO Feature lifting

Our approach to lift 2D masks to 3D Gaussian Splatting fields can also lift any 2D foundation model features onto 3D Gaussians using the same strategy. Consequently, we lift DINOv2 [34] features onto a 3DGS field using our LBG approach. We visualize the first 24 PCA components of the features in Figure 12.

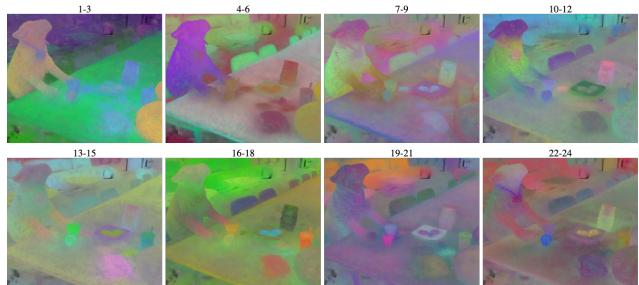


Figure 12. **Lifting DINOv2 features onto Gaussians.** Using our Lifting-by-Gaussians approach, we lift DINOv2 features and visualize the first 24 PCA components.