The Mask Is Not the Object: Volumetric Supervision for 3D Gaussian Segmentation

Jeonghwan Cho **Minsu Kim** **Junyoung Hong** **Seon Joo Kim**

Yonsei University

Project Page

Abstract

3D Gaussian Splatting enables object-level operations on Gaussian primitives, yet existing segmentation methods are largely trained and evaluated through rendered views. This mismatch can produce accurate rendered masks from incomplete 3D selections since alpha-blended feature supervision concentrates learning on dominant contributors while leaving occluded or low-contribution Gaussians weakly supervised. We propose VODA, a plug-and-play supervision objective that exploits local depth structure to group per-pixel contributors and directly supervises all Gaussians in selected depth clusters, bypassing contribution-weighted gradient delivery. To evaluate volumetric completeness, we introduce Removal-3D, a synthetic object-removal benchmark, and the Removal Depth Error (RDE), a metric comparing predicted and ground-truth depth changes. Across four rendering-based baselines, VODA consistently improves volumetric segmentation on Removal-3D while maintaining or improving performance on LERF and ScanNet. Further analysis shows that VODA recovers lower-contribution Gaussians, while simple gradient reweighting and spatial propagation fail to reproduce comparable gains.

1 Introduction

Object-level editing of reconstructed scenes requires selecting objects as complete 3D entities. Removing, relocating, or extracting an object depends on capturing its full spatial extent, including its interior. For scenes represented by 3D Gaussian Splatting (3DGS) [12], this means selecting all Gaussians associated with the object. Yet existing segmentation methods [22, 24, 29, 10, 28, 6] are commonly evaluated by the accuracy of their rendered 2D masks, which does not establish whether the underlying 3D selection is complete. Figure 1 illustrates this gap: a selection can produce an accurate mask yet miss some of the object's Gaussians, leaving visible remnants after removal.

This gap stems from rendering-based feature supervision. Existing methods [29, 10, 28, 6] learn per-Gaussian instance features through rendered 2D feature maps, using external feature or mask targets, or mask-based contrastive objectives. Despite their differences, these objectives deliver gradients according to each Gaussian's rendering contribution (§3.1). Dominant contributors can thus suffice to render an accurate mask, while low-contribution Gaussians belonging to the same object receive weak or no supervision and may remain unselected.

We therefore study *volumetric 3D Gaussian segmentation*, aiming to select object-associated Gaussians beyond those needed to render a correct mask. We propose **VODA**, a plug-and-play objective that extends object-level supervision beyond the dominant contributors. VODA modifies only feature supervision, leaving the baseline's reconstructed scene, feature representation, and inference-time selection procedure unchanged.

Our key observation is that, after RGB optimization, Gaussians associated with the same object tend to cluster at similar depths, while larger depth gaps often reflect separation between distinct

40th Conference on Neural Information Processing Systems (NeurIPS 2026).

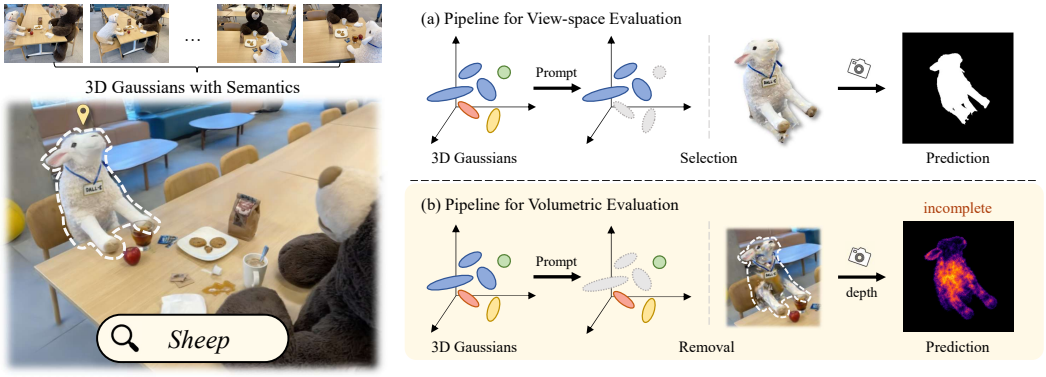

Figure 1: **A plausible rendered mask does not imply complete 3D selection.** For the query "sheep," (a) the selected Gaussians render a plausible object mask, but (b) the depth change after removing the same Gaussian set indicates incomplete object removal in the reconstructed scene. This exposes a gap between rendered-mask quality and Gaussian-level selection completeness.

objects. VODA exploits this structure by grouping per-pixel contributors into depth clusters and selecting the cluster that best explains each pixel (§3.2). Targets constructed from rendered features are then applied directly to every Gaussian in the selected cluster, supervising both dominant and low-contribution members while bypassing the contribution-weighted gradient path (§3.3). This direct supervision extends coverage within objects, while a complementary pixel-level push loss sharpens boundaries between adjacent objects.

Evaluating volumetric segmentation requires measuring whether an object selection is complete beyond its visible projection. Existing benchmarks score rendered masks or visible surface labels and therefore do not directly measure this property. We use object removal as a diagnostic: removing a complete selection should produce the same depth change as removing the target object itself. We therefore introduce Removal-3D, a synthetic indoor-scene benchmark with paired ground-truth depth maps before and after target removal. To quantify how closely a predicted selection reproduces this change, we propose Removal Depth Error (RDE), which compares the ground-truth depth change with the change induced by removing the predicted Gaussians.

Experiments across four rendering-based baselines [28, 6, 29, 10] show that VODA consistently improves volumetric segmentation on Removal-3D while maintaining or improving performance on standard benchmarks [13, 9]. Ablations identify direct Gaussian-level supervision as the primary source of these gains, while additional diagnostics support the need to change which Gaussians receive supervision rather than merely modifying the existing rendering-based signal.

In summary, our contributions are threefold. (1) We formulate volumetric 3D Gaussian segmentation, distinguishing selection completeness from rendered-mask accuracy and identifying the bias in rendering-based supervision. (2) We propose VODA, a plug-and-play objective that directly supervises all Gaussians in selected depth clusters. (3) We introduce Removal-3D and RDE, a benchmark and metric that assess selection completeness through removal-induced depth changes.

2 Related Work

The explicit primitives of 3D Gaussian Splatting (3DGS) [12] support object-level selection and editing. Recent methods attach semantic features or object identities to Gaussians for object-level scene understanding [29, 33]. We focus on how 2D supervision is transferred to Gaussians, distinguishing rendering-based feature learning from direct 2D-to-Gaussian association.

Rendering-based Feature Learning. Earlier NeRF methods lift 2D semantics through feature distillation, mask supervision, or contrastive learning [31, 16, 26, 25, 1, 19, 4, 14, 30, 2], including distillation of pretrained visual features such as DINO [3]. Rendering-based 3DGS methods similarly alpha-blend per-Gaussian features into 2D maps, with two main supervision paradigms. **Supervised feature learning** matches these maps to external targets: CLIP features [22, 24, 32, 34, 23, 18], SAM mask identities as in Gaussian Grouping [29], or both jointly [10]. **Self-supervised feature learning** instead uses SAM masks as grouping cues, pulling features within a mask together and pushing features across masks apart without explicit feature targets [28, 6, 5, 8, 17]. Although their

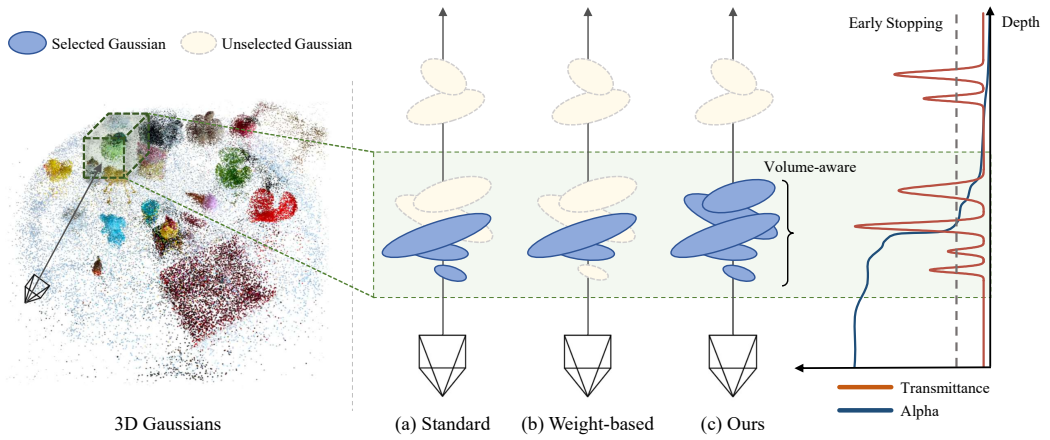

Figure 2: **Comparison of supervision and association strategies for per-pixel Gaussian contributors.** (a) Rendering-based feature learning delivers contribution-weighted gradients, giving weak supervision to low-contribution Gaussians. (b) Direct association links each pixel to a sparse subset of contributors selected by contribution-based rules. (c) VODA directly supervises every Gaussian in the selected depth cluster, including low-contribution Gaussians associated with the same object.

supervision sources differ, these rendering-based objectives share the contribution-weighted gradient path analyzed in §3.1. As illustrated in Figure 2a, low-contribution Gaussians can therefore receive weak supervision even when the rendered features match the 2D targets or grouping constraints. VODA complements either paradigm with direct Gaussian-level supervision while leaving the reconstructed scene, feature representation, and inference-time selection unchanged.

Direct 2D-to-Gaussian Association. Another line of work associates 2D observations directly with Gaussians rather than learning features through rendered-map losses. Lifting By Gaussians [7] assigns 2D mask labels to the per-pixel maximum-contributing Gaussian and merges the resulting fragments across views. Dr. Splat [15] registers CLIP embeddings to dominant or Top-k Gaussians for optimization-free 3D search. Quantile Rendering [11] uses quantile-based sampling to associate high-dimensional features with a sparse set of influential Gaussians. Despite their different association rules, these methods associate each pixel with only a sparse subset of Gaussian contributors, as illustrated in Figure 2b. Multi-view observations can expand this set, but object-associated Gaussians that remain weak contributors across views may still receive no assignment. Thus, bypassing rendered-feature optimization alone does not necessarily resolve the incompleteness induced by sparse, contribution-based association. VODA instead uses per-pixel depth structure to identify the target-associated cluster and directly supervises all Gaussians within it.

3 Method

We propose VODA, a plug-and-play supervision objective for volumetric 3D Gaussian segmentation. We first analyze why standard rendering-based supervision is insufficient in §3.1. The depth-aware rasterizer in §3.2 then groups per-pixel contributors and selects a target depth cluster. Direct Gaussian-level supervision in §3.3 applies a feature loss to every Gaussian in the selected cluster, together with a complementary pixel-level push loss that encourages separation between object instances.

3.1 Contribution-weighted Supervision in Feature Rendering

Let $\{g_i\}_{i=1}^{N}$ be the Gaussians of a reconstructed 3DGS scene, and let $\mathbf{f}_i \in \mathbb{R}^C$ be the learnable instance feature associated with Gaussian g_i. For a pixel p, the Gaussians overlapping the pixel are sorted by camera-space depth and alpha-blended into a rendered feature map:

$$\mathbf{F}(p) = \sum_{i \in \mathcal{G}(p)} w_i(p)\mathbf{f}_i, \qquad w_i(p) = T_i(p)\alpha_i(p), \tag{1}$$

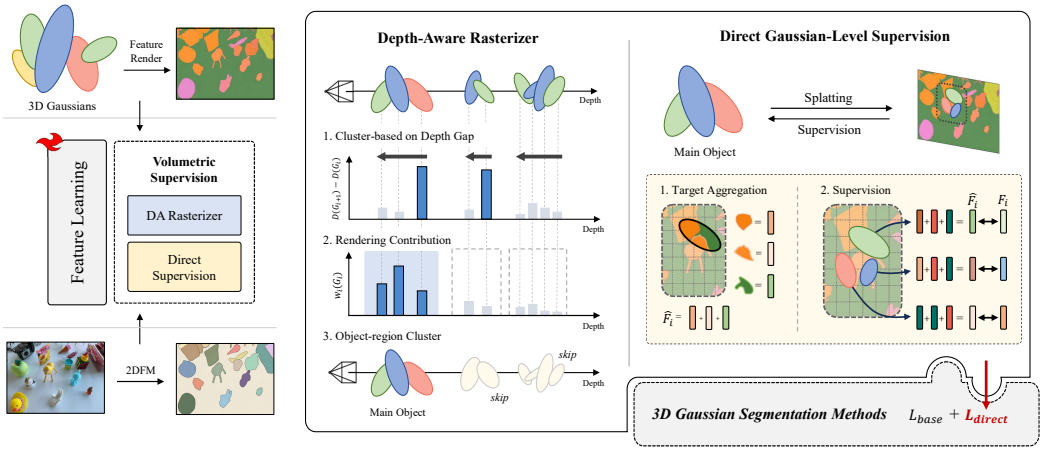

Figure 3: **Overview of VODA.** The depth-aware rasterizer groups per-pixel contributors using adjacent depth gaps and selects the cluster with the largest total rendering contribution. Alpha-weighted pixel features are aggregated into detached Gaussian-level targets. A direct feature loss supervises every Gaussian in the selected cluster, including low-contribution Gaussians, without contribution-weighted gradient delivery. The baseline feature-rendering loss remains active.

where $\mathcal{G}(p)$ denotes the Gaussians contributing to pixel p in depth order, $\alpha_i(p)$ is the per-pixel alpha value of g_i, and $T_i(p)$ is the accumulated transmittance before g_i in that ordering.

Let $\mathcal{L}_{\text{rend}}$ denote the component of the baseline feature-learning objective that is computed through the rendered feature maps. For this rendered-map supervision term, applying the chain rule gives:

$$\frac{\partial \mathcal{L}_{\text{rend}}}{\partial \mathbf{f}_i} = \sum_{p: i \in \mathcal{G}(p)} w_i(p) \frac{\partial \mathcal{L}_{\text{rend}}}{\partial \mathbf{F}(p)}. \tag{2}$$

This contribution-weighted path is shared by both paradigms described in §2, whether rendered features are matched to external targets or compared across pixels. As Eq. 2 shows, this shared path leads to an imbalance in gradient delivery: dominant contributors can learn features sufficient for an accurate rendered mask, while low-contribution Gaussians associated with the same object receive weak gradients due to their small rendering weights.

3.2 Depth-Aware Rasterizer

To address this imbalance, we need to identify object-associated Gaussians beyond those retained by maximum or Top-k selection (Figure 2). Instead of relying solely on individual rendering contributions, we exploit the local depth structure of the reconstructed scene. During reconstruction, cloning and splitting introduce new Gaussians around existing primitives [12]. This local refinement motivates using depth proximity as a cue for object association: contributors representing the same local object region can form a depth cluster containing both dominant and low-contribution Gaussians. Larger gaps can indicate separation from other objects or the background.

Depth clustering. To group contributors based on these depth gaps, we introduce a depth-aware rasterizer that collects per-pixel contributors, their camera-space depths, and alpha values. It disables transmittance-based early termination to retain contributors behind dominant Gaussians. For each pixel p with at least two contributors, let $d_j(p)$ denote the camera-space depth of the j-th contributor in depth order. We split consecutive contributors whenever their depth gap exceeds $\tau(p)$:

$$\tau(p) = \text{clip}\left(\gamma \cdot \text{median}_{1 \le j < |\mathcal{G}(p)|} \left[d_{j+1}(p) - d_j(p)\right], \tau_{\min}, \tau_{\max}\right). \tag{3}$$

The median adapts the threshold to the local gap distribution, γ controls split sensitivity, and clipping bounds it within $[\tau_{\min}, \tau_{\max}]$. We denote the resulting depth clusters by $\mathcal{C}_k(p)$.

Target cluster selection. To identify the target cluster, VODA selects the cluster with the largest total contribution at each pixel:

$$\mathcal{C}^*(p) = \arg\max_{\mathcal{C}_k(p)} \sum_{i \in \mathcal{C}_k(p)} w_i(p). \tag{4}$$

Using total contribution makes cluster selection less sensitive to near-camera floater Gaussians than relying on proximity alone.

3.3 Direct Gaussian-Level Supervision

Given the selected depth cluster, we directly supervise its Gaussian features to bypass the contribution-weighted rendering path. Since there are no ground-truth instance features for individual Gaussians, we construct Gaussian-level pseudo-targets from the full rendered feature $\mathbf{F}(p)$ defined in Eq. 1, rather than rendering the selected cluster alone. This feature is dominated by high-contribution Gaussians that receive strong baseline supervision, providing a learned feature signal for other Gaussians in the same cluster. We detach the resulting Gaussian-level targets with $\mathrm{sg}[\cdot]$ and apply the loss directly to Gaussian features, while retaining the baseline's original rendering-based supervision.

Gaussian-level target aggregation. A Gaussian may be selected at multiple pixels, which we collect as $\mathcal{P}_i = \{p \mid g_i \in \mathcal{C}^*(p)\}$. This set contains only pixels where g_i is selected, not every pixel to which it contributes. VODA aggregates the rendered features over $\mathcal{P}_i$ to obtain $\hat{\mathbf{F}}_i$:

$$\hat{\mathbf{F}}_i = \mathrm{sg}\left[\frac{\sum_{p \in \mathcal{P}_i} \alpha_i(p)\mathbf{F}(p)}{\sum_{p \in \mathcal{P}_i} \alpha_i(p) + \epsilon}\right], \tag{5}$$

where ϵ ensures numerical stability and $\alpha_i(p)$ weights each target by the Gaussian's alpha at p.

Direct feature loss. The direct loss aligns selected Gaussian features with aggregated targets:

$$\mathcal{L}_{\mathrm{direct}} = \frac{1}{|\mathcal{G}_{\mathrm{sel}}|} \sum_{g_i \in \mathcal{G}_{\mathrm{sel}}} \left\| \mathbf{f}_i - \hat{\mathbf{F}}_i \right\|_2^2, \qquad \mathcal{G}_{\mathrm{sel}} = \{g_i \mid |\mathcal{P}_i| > 0\}. \tag{6}$$

The loss acts directly on Gaussian features with detached targets, so its gradient bypasses rendering weights and reaches low-contribution Gaussians in the selected clusters. Eq. 6 uses squared-ℓ_2 by default, while baseline-specific distance choices are provided in the supplementary.

Pixel-level feature push. A depth cluster can span adjacent object instances, so we add a pixel-level push loss to reduce feature mixing across object boundaries. Within each training view, we randomly sample a set $\mathcal{Q}$ of pixel pairs (p, q) belonging to different non-background masks and penalize pairs whose rendered-feature cosine similarity exceeds a margin m in the learned feature space:

$$\mathcal{L}_{\mathrm{push}} = \frac{1}{|\mathcal{Q}|} \sum_{(p,q) \in \mathcal{Q}} \max(0,\, \cos(\mathbf{F}(p), \mathbf{F}(q)) - m)\,. \tag{7}$$

3.4 Training Objective

Let $\mathcal{L}_{\mathrm{base}}$ denote the baseline feature-learning objective, including any existing Gaussian-level regularization. We retain this objective and add VODA's direct and pixel-level push losses:

$$\mathcal{L}_{\mathrm{VODA}} = \lambda_{\mathrm{direct}}\mathcal{L}_{\mathrm{direct}} + \lambda_{\mathrm{push}}\mathcal{L}_{\mathrm{push}}, \qquad \mathcal{L} = \mathcal{L}_{\mathrm{base}} + \mathcal{L}_{\mathrm{VODA}}. \tag{8}$$

Depending on the baseline, VODA is applied during feature learning on a reconstructed scene or in a post-training feature-only fine-tuning stage. Scene geometry, opacity, feature representation, and inference-time selection remain unchanged, and VODA adds no parameters or computation at inference. Implementation schedules and hyperparameters are provided in the supplementary.

Table 1: Open-vocabulary 3D object selection on LERF under the select-then-render protocol. We report rendered-mask mIoU and mAcc@0.25. `waldo_kitchen` is abbreviated as `kitchen`.

Methods	mIoU ↑					mAcc. ↑				
	figurines	teatime	ramen	kitchen	**Mean**	figurines	teatime	ramen	kitchen	**Mean**
LangSplat [22]	10.16	11.38	7.92	9.18	9.66	8.93	20.34	11.27	9.09	12.41
LEGaussians [24]	17.99	19.27	15.79	11.78	16.21	23.21	27.12	26.76	18.18	23.82
OpenGaussian [28]	49.19	45.63	22.34	**31.33**	37.12	**69.64**	74.58	32.39	**50.00**	56.65
+ VODA	**55.31**	**62.89**	**25.26**	28.47	**42.98**	**69.64**	**79.66**	**36.62**	45.45	**57.84**
LaGa [6]	**65.79**	60.59	**53.35**	47.47	56.80	**89.29**	83.05	64.79	68.18	76.33
+ VODA	63.28	**69.05**	50.42	**58.38**	**60.28**	**89.29**	**91.53**	**67.61**	**81.82**	**82.56**
Gaussian Grouping [29]	26.04	26.10	16.10	—	22.75	46.43	46.67	33.33	—	42.14
+ VODA	**51.45**	**43.64**	**36.09**	—	**43.73**	**82.14**	**80.00**	**66.67**	—	**76.27**
ILGS [10]	24.43	38.70	26.87	—	30.00	50.00	75.00	55.56	—	60.19
+ VODA	**42.36**	**58.32**	**48.19**	—	**49.62**	**75.00**	**91.67**	**77.78**	—	**81.48**

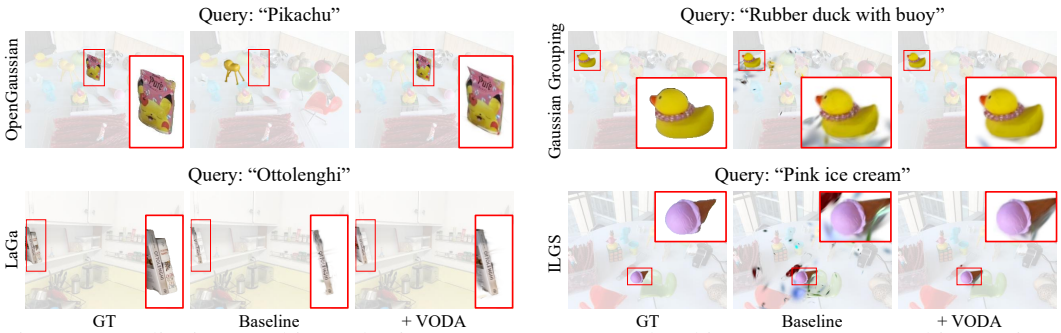

Figure 4: Qualitative 3D object selection on LERF. For "Ottolenghi," VODA recovers object regions missed by the baselines, while for "Pink ice cream," it suppresses spurious selections. These examples demonstrate improvements in both the completeness and precision of 3D Gaussian selection.

4 Experiments

We evaluate VODA on two standard 3DGS segmentation benchmarks, LERF and ScanNet, in §4.1, and on our Removal-3D benchmark for volumetric evaluation in §4.2. We further ablate its design choices in §4.3 and analyze recovered Gaussians and contribution-based alternatives in §4.4. We evaluate four baselines spanning both feature-learning paradigms in §2: OpenGaussian [28] and LaGa [6] are self-supervised, while Gaussian Grouping [29] and ILGS [10] are supervised.

Implementation details. We implement VODA on the official codebase of each baseline and apply it only to the feature-learning stage, leaving RGB reconstruction and inference unchanged. The depth-clustering and loss hyperparameters are shared across all four baselines. VODA is applied within the existing feature stage for OpenGaussian and LaGa, and as a short feature-only fine-tuning stage for Gaussian Grouping and ILGS. Full implementation details are provided in the supplementary.

4.1 Comparison on Standard Benchmarks

We evaluate VODA on two complementary standard settings probing different aspects of 3D scene understanding: open-vocabulary 3D object selection on LERF and semantic segmentation on ScanNet.

3D Object Selection on LERF. Following OpenGaussian, we perform open-vocabulary 3D object selection by selecting a set of Gaussians for each text query and rendering only that set into multiple views. We compare these rendered masks with the ground-truth object masks from LangSplat [22], reporting mIoU and the percentage of queries whose IoU exceeds 0.25 (mAcc@0.25). Because selection precedes rendering, this protocol evaluates a single 3D Gaussian selection consistently across views rather than independently segmenting each rendered image. Gaussian Grouping and

Table 2: Text-query 3D semantic segmentation on ScanNet [9]. We report point-cloud mIoU and mAcc for three category splits. Gaussian Grouping and ILGS are omitted (see text).

Methods	19 classes		15 classes		10 classes	
	mIoU ↑	mAcc. ↑	mIoU ↑	mAcc. ↑	mIoU ↑	mAcc. ↑
LangSplat [22]	3.78	9.11	5.35	13.20	8.40	22.06
LEGaussians [24]	3.84	10.87	9.01	22.22	12.82	28.62
OpenGaussian [28]	26.20	39.35	28.00	42.67	34.57	48.38
+ VODA	**29.00**	**42.80**	**29.43**	**44.38**	**36.89**	**50.69**
LaGa [6]	32.20	47.35	37.19	53.88	43.06	61.31
+ VODA	**33.20**	**49.63**	**37.21**	**55.59**	**45.53**	**65.24**

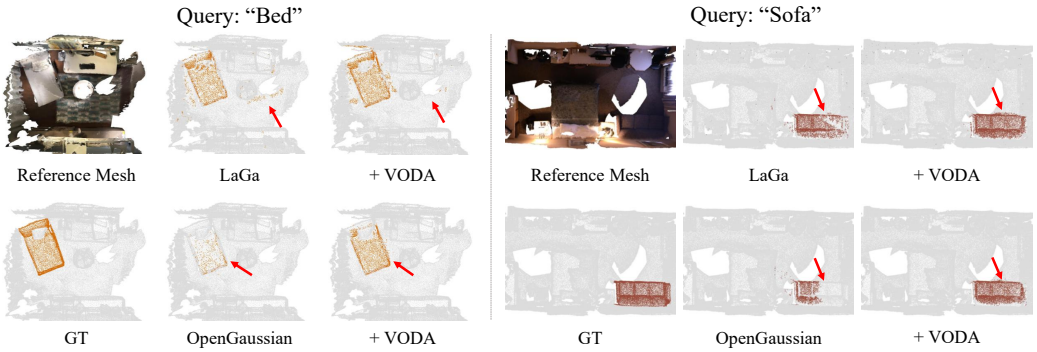

Figure 5: Qualitative 3D semantic segmentation on ScanNet. VODA recovers missing object regions and suppresses spurious semantic assignments for both "Bed" and "Sofa." Red arrows highlight representative regions with incomplete or noisy baseline predictions corrected by VODA.

ILGS originally produce 2D segmentations by rendering first and then grouping the rendered features. We instead evaluate them with the same select-then-render protocol as OpenGaussian and LaGa, so their results differ from those reported in the original papers. Table 1 shows that VODA improves the average mIoU and mAcc@0.25 across all four baselines. The gains are particularly large for Gaussian Grouping and ILGS, although a few individual scenes show slight regressions. Figure 4 shows that VODA recovers missing object parts while reducing spurious selections.

3D Semantic Segmentation on ScanNet. Following OpenGaussian, we use 19, 15, and 10 ScanNetv2 [9] categories as text queries and assign each Gaussian to its most similar query. We transfer these predictions to the ScanNet point cloud and report mIoU and mAcc against the ground-truth semantic labels. ScanNet complements LERF by evaluating category-level semantic assignment over the entire scene rather than object-level selection from individual text queries. Table 2 shows consistent improvements for both OpenGaussian and LaGa across all three category splits as the semantic label set changes from 19 to 10 classes. The gains remain consistent across different category splits, suggesting that VODA is not tied to a particular semantic class partition. Gaussian Grouping and ILGS are not applicable because their features are tied to per-view SAM mask identities, which the ScanNet text-query protocol does not provide. Figure 5 shows that VODA produces cleaner semantic assignments around object boundaries, with less label leakage across neighboring regions. However, ScanNet ground-truth labels cover only visible surfaces in the reconstructed point cloud and therefore cannot directly assess occluded object-associated Gaussians, motivating our volumetric benchmark.

4.2 Comparison on Volumetric Benchmark

Removal-3D. Standard benchmarks evaluate rendered masks or visible surface labels, but cannot determine whether the selected set covers all Gaussians associated with the 3D object. In particular, they do not test whether removing the selected Gaussians reveals the surfaces expected after removing the object itself. We construct Removal-3D, a synthetic benchmark with three indoor scenes (Table,

Table 3: Volumetric segmentation on Removal-3D. Rendered-mask mIoU measures spatial overlap, while mRDE measures the depth discrepancy after removing the selected Gaussians.

Methods	Table		Kitchen		Room		**Mean**	
	mIoU↑	mRDE↓	mIoU↑	mRDE↓	mIoU↑	mRDE↓	mIoU↑	mRDE↓
OpenGaussian [28]	57.87	43.31	37.46	29.05	63.26	45.56	52.87	39.31
+ VODA	**65.76**	**29.39**	**52.38**	**13.00**	**64.55**	**39.64**	**60.90**	**27.34**
LaGa [6]	77.02	43.95	66.23	22.99	84.55	43.85	75.93	36.93
+ VODA	**79.79**	**32.25**	**70.87**	**8.16**	**86.54**	**24.56**	**79.07**	**21.66**
Gaussian Grouping [29]	25.57	52.65	18.52	33.92	34.59	63.57	26.23	50.05
+ VODA	**35.94**	**41.59**	**18.66**	**20.29**	**35.93**	**52.47**	**30.18**	**38.12**
ILGS [10]	27.14	50.76	26.94	32.86	40.88	66.31	31.66	49.98
+ VODA	**37.24**	**45.19**	**31.46**	**18.40**	**47.54**	**43.34**	**38.75**	**35.65**

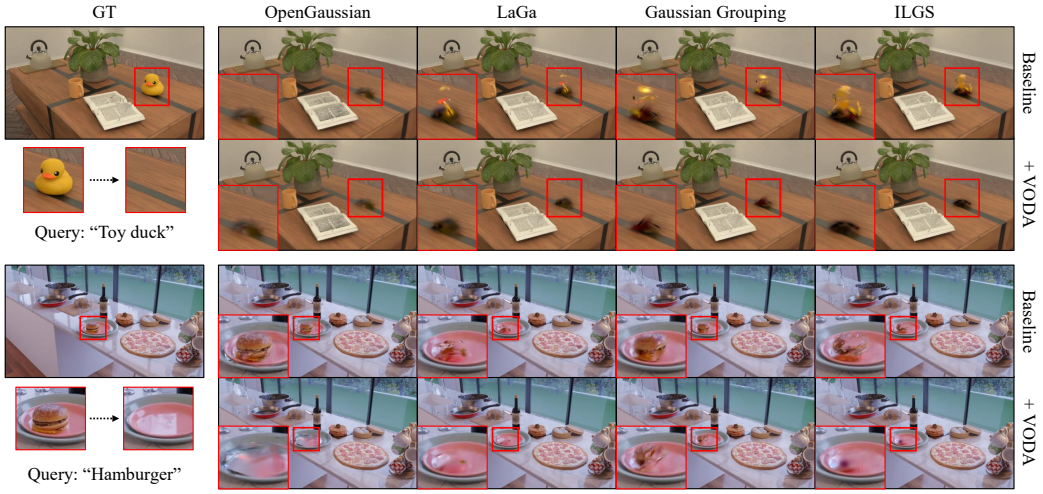

Figure 6: Qualitative removal on Removal-3D. Each scene is rendered after removing the queried Gaussians. VODA leaves less residual target content and better reveals the surface behind the object.

Kitchen, and Room) built in Blender using BlenderKit assets. For each target object, the benchmark provides paired depth maps of the full scene and the scene after ground-truth object removal.

Removal Depth Error. We introduce the Removal Depth Error (RDE) to compare this reference depth change with the change induced by removing the predicted Gaussian set:

$$\mathrm{RDE} = \frac{1}{|\Omega|} \sum_{p \in \Omega} \rho(\Delta D(p),\, \Delta D_{\mathrm{gt}}(p))\,, \tag{9}$$

where Ω denotes the ground-truth target mask region, and $\Delta D(p)$ and $\Delta D_{\mathrm{gt}}(p)$ are the predicted and reference depth changes, respectively. The function ρ clips the predicted change to the range bounded by the magnitude of the reference change before computing absolute error. We obtain per-scene mRDE by averaging over viewpoints and target objects, and report the mean across the three scenes as the overall score. Because RDE is evaluated only within the target object mask, selections outside this region do not affect the score. We therefore report rendered-mask mIoU alongside RDE as a complementary measure. RDE measures whether the selected Gaussians remove the target volume as expected, while mIoU measures whether the selection remains spatially aligned with the target in rendered views. The complete computation is provided in the supplementary.

Volumetric Evaluation on Removal-3D. Table 3 shows consistent reductions in mRDE for every baseline and scene, together with improved mIoU. The improvement holds across both feature-learning paradigms, demonstrating improved Gaussian-level selection completeness on Removal-3D, a property standard benchmarks leave unmeasured. The mIoU gains suggest that lower mRDE is not achieved by simply expanding the selected set. Figure 6 complements the LERF and ScanNet

visualizations by rendering each scene after removing the queried Gaussians. For the "toy duck" and "hamburger" queries, the baselines leave residual target content, indicating that some object Gaussians remain unselected. VODA leaves less residual content and better reveals the occluded surface.

4.3 Ablation Study

We ablate supervision delivery, the pixel-level push, and target aggregation using OpenGaussian and Gaussian Grouping as representatives of self-supervised and supervised feature learning.

Table 4: Ablation on supervision type and pixel-level push.

Supervision		Pixel-Push	LERF		Removal-3D	
Rendering	Direct		mIoU↑	mAcc↑	mIoU↑	mRDE↓
OpenGaussian						
✓		✓	39.06	56.18	52.15	32.37
	✓		42.57	**58.14**	57.90	32.63
	✓	✓	**42.98**	57.84	**60.90**	**27.34**
Gaussian Grouping						
✓		✓	12.40	13.28	16.84	55.60
	✓		42.89	**76.27**	29.57	38.25
	✓	✓	**43.73**	**76.27**	**30.18**	**38.12**

Table 5: Target aggregation.

Aggregation		LERF	
Simple	Alpha	mIoU↑	mAcc↑
OpenGaussian			
✓		40.63	55.89
	✓	**42.98**	**57.84**
Gaussian Grouping			
✓		43.22	**76.27**
	✓	**43.73**	**76.27**

Supervision delivery. We compare VODA's direct loss with a rendering variant that uses the same selected depth clusters and target features, but delivers supervision through feature rendering. Thus, the two variants differ only in how supervision reaches the selected Gaussians. Direct supervision consistently performs better across both baselines on LERF and Removal-3D, with the largest difference on Gaussian Grouping, where LERF mIoU rises from 12.40 to 43.73. This shows that identifying the appropriate depth cluster is not sufficient. Supervision must also bypass the contribution-weighted rendering path and reach the selected Gaussian features directly.

Pixel-level push. We next isolate the pixel-level push loss while retaining direct supervision. The push loss has little effect on standard LERF performance, but consistently improves volumetric selection on Removal-3D. This supports its complementary role to direct supervision. Direct supervision improves Gaussian coverage, while the push loss reduces feature mixing between nearby instances and improves volumetric selection. Boundary visualizations in the supplementary illustrate this effect.

Target aggregation. Finally, we compare uniform averaging with the α-weighted aggregation in Eq. 5. Alpha weighting yields modest but consistent improvements across both baselines, supporting the choice to place greater trust in pixels where the selected Gaussian has a higher per-pixel alpha. Overall, the ablation shows that direct supervision provides the primary benefit, while the pixel-level push improves instance separation and α weighting improves target reliability.

4.4 Analysis

We next test whether the Gaussians additionally recovered by VODA are precisely those that receive weak supervision through the rendering path, and whether simpler reweighting or spatial propagation can address the same failure mode.

Rendering contribution of recovered Gaussians. Using LaGa, we compare Gaussians retrieved by the baseline with those missed by the baseline but recovered after adding VODA. For each Gaussian, we record its maximum rendering contribution $w_i(p)$ over all training pixels and views. Because the baseline and VODA checkpoints share identical geometry and opacity, as detailed in the supplementary, these contribution statistics are measured under the same scene geometry. Figure 7 shows that the recovered Gaussians are consistently shifted toward lower rendering contribution on Removal-3D. At the $1/255$ cutoff, the fraction of low-contribution Gaussians is 10.2–15.9 percentage points higher in the recovered set across the three scenes. The supplementary extends this analysis to

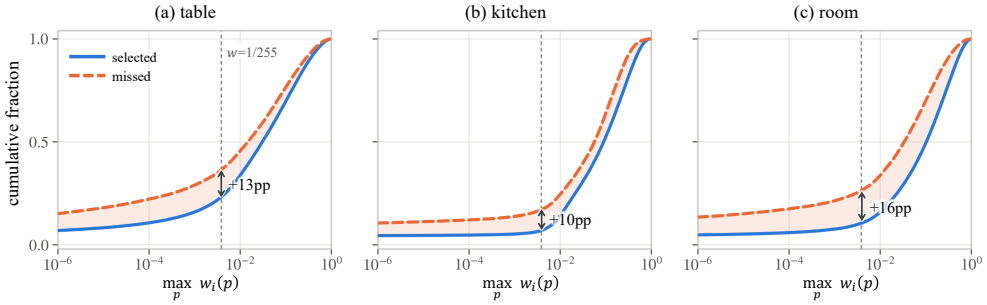

Figure 7: Rendering-contribution CDFs for LaGa on Removal-3D. VODA-recovered Gaussians have lower contributions than baseline-selected ones; $1/255$ marks the reference threshold.

Table 6: Contribution-based alternatives. Reweighting and propagation yield inconsistent gains, while VODA improves all three metrics for both baselines through direct depth-cluster supervision.

OpenGaussian				*Gaussian Grouping*			
Methods	LERF	Removal-3D		Methods	LERF	Removal-3D	
	mIoU ↑	mIoU ↑	mRDE ↓		mIoU ↑	mIoU ↑	mRDE ↓
Baseline	37.12	52.87	39.3	Baseline	22.75	26.23	50.0
Normalize	24.61	17.1	72.7	Normalize	34.03	1.6	48.7
Clamp	35.94	13.4	69.4	Clamp	31.42	1.5	52.2
KNN	30.28	17.9	73.3	KNN	12.24	1.5	54.7
VODA	**42.98**	**60.90**	**27.3**	VODA	**43.73**	**30.18**	**38.1**

six scenes, with accumulated-contribution statistics showing the same trend. This directly supports our diagnosis that VODA preferentially recovers Gaussians with weaker baseline supervision.

Contribution reweighting and spatial propagation. Table 6 tests whether the imbalance can be addressed without explicitly assigning supervision through depth clusters. *Normalize* and *Clamp* rescale the rendering gradient, while *KNN* propagates features from nearby high-contribution Gaussians. None of these alternatives matches VODA. All three substantially degrade volumetric performance for OpenGaussian, while for Gaussian Grouping, Normalize improves LERF performance but severely degrades Removal-3D. These failures reflect a limitation of contribution-based heuristics. Rendering contribution alone does not indicate whether a Gaussian is associated with the target object. Reweighting therefore cannot identify the relevant low-contribution Gaussians or recover those that receive no rendered gradient. Indeed, across the six analyzed scenes, 8.5–18.7% of recovered Gaussians receive zero contribution in every training view. KNN avoids this zero-gradient issue, but spatial proximity likewise does not establish object association and may propagate features across object boundaries.

5 Conclusion

VODA closes the gap between rendered-mask accuracy and volumetric completeness by grouping contributors by depth and directly supervising every Gaussian in selected clusters. We introduce Removal-3D and RDE to measure selection completeness from depth changes after object removal. Across four baselines, VODA improves applicable standard benchmarks and both mRDE and rendered-mask mIoU on Removal-3D. Reweighting and spatial propagation fail to match VODA, suggesting that reaching low-contribution Gaussians matters more than scaling rendered gradients.

Limitations. Depth clustering may merge contacting objects or fragment objects with sparse interiors, and the pixel-level push only partly addresses these cases. Because occluded Gaussians lack feature-level ground truth, we assess their supervision indirectly through downstream selection and removal. Exact volumetric evaluation requires paired full-scene and object-removed depth maps, available only for the three synthetic Removal-3D scenes.

Acknowledgments

This work was supported by the Artificial Intelligence Graduate School Program, Yonsei University (RS-2020-II201361), Institute of Information & communications Technology Planning & Evaluation (IITP) grants funded by the Korea government (MSIT) (RS-2024-00457882) and the Global AI Frontier Lab International Collaborative Research (RS-2024-00469482 & RS-2024-00509279).

References

[1] Yash Bhalgat, Iro Laina, João F. Henriques, Andrew Zisserman, and Andrea Vedaldi. Contrastive lift: 3D object instance segmentation by slow-fast contrastive fusion. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2023.

[2] Yash Bhalgat, Iro Laina, João F. Henriques, Andrew Zisserman, and Andrea Vedaldi. N2F2: Hierarchical scene understanding with nested neural feature fields. In *European Conference on Computer Vision (ECCV)*, 2024.

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

[4] Jiazhong Cen, Zanwei Zhou, Jiemin Fang, Chen Yang, Wei Shen, Lingxi Xie, Dongsheng Jiang, Xiaopeng Zhang, and Qi Tian. Segment anything in 3D with NeRFs. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2023.

[5] Jiazhong Cen, Jiemin Fang, Chen Yang, Lingxi Xie, Xiaopeng Zhang, Wei Shen, and Qi Tian. Segment any 3D Gaussians. In *AAAI Conference on Artificial Intelligence (AAAI)*, 2025.

[6] Jiazhong Cen, Xudong Zhou, Jiemin Fang, Changsong Wen, Lingxi Xie, Xiaopeng Zhang, Wei Shen, and Qi Tian. Tackling view-dependent semantics in 3D language Gaussian splatting. In *International Conference on Machine Learning (ICML)*, 2025.

[7] Rohan Chacko, Nicolai Häni, Eldar Khaliullin, Lin Sun, and Douglas Lee. Lifting by Gaussians: A simple, fast and flexible method for 3D instance segmentation. In *IEEE/CVF Winter Conference on Applications of Computer Vision (WACV)*, 2025.

[8] Seokhun Choi, Hyeonseop Song, Jaechul Kim, Taehyeong Kim, and Hoseok Do. Click-Gaussian: Interactive segmentation to any 3D Gaussians. In *European Conference on Computer Vision (ECCV)*, 2024.

[9] Angela Dai, Angel X. Chang, Manolis Savva, Maciej Halber, Thomas Funkhouser, and Matthias Nießner. ScanNet: Richly-annotated 3D reconstructions of indoor scenes. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017.

[10] SungMin Jang and Wonjun Kim. Identity-aware language gaussian splatting for open-vocabulary 3d semantic segmentation. In *Proceedings of the International Conference on Computer Vision*, 2025.

[11] Yoonwoo Jeong, Cheng Sun, Frank Wang, Minsu Cho, and Jaesung Choe. Quantile rendering: Efficiently embedding high-dimensional feature on 3D Gaussian splatting. *arXiv preprint arXiv:2512.20927*, 2025.

[12] Bernhard Kerbl, Georgios Kopanas, Thomas Leimkühler, and George Drettakis. 3D Gaussian Splatting for real-time radiance field rendering. *ACM Transactions on Graphics*, 42(4), 2023.

[13] Justin Kerr, Chung Min Kim, Ken Goldberg, Angjoo Kanazawa, and Matthew Tancik. LERF: Language embedded radiance fields. In *IEEE/CVF International Conference on Computer Vision (ICCV)*, 2023.

[14] Chung Min Kim, Mingxuan Wu, Justin Kerr, Ken Goldberg, Matthew Tancik, and Angjoo Kanazawa. GARField: Group anything with radiance fields. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2024.

[15] Jun-Seong Kim, GeonU Kim, Yu-Ji Kim, Yu-Chiang Frank Wang, Jaesung Choe, and Tae-Hyun Oh. Dr. Splat: Directly referring 3D Gaussian splatting via direct language embedding registration. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2025.

[16] Sosuke Kobayashi, Eiichi Matsumoto, and Vincent Sitzmann. Decomposing NeRF for editing via feature field distillation. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2022.

[17] Hyunjee Lee, Youngsik Yun, Jeongmin Bae, Seoha Kim, and Youngjung Uh. Rethinking open-vocabulary segmentation of radiance fields in 3D space. In *AAAI Conference on Artificial Intelligence (AAAI)*, 2025.

[18] Hao Li, Roy Qin, Zhengyu Zou, Diqi He, Bohan Li, Bingquan Dai, Dingwen Zhang, and Junwei Han. LangSurf: Language-embedded surface Gaussians for 3D scene understanding. *arXiv preprint arXiv:2412.17635*, 2024.

[19] 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. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2023.

[20] Yiren Lu, Yunlai Zhou, Yiran Qiao, Chaoda Song, Tuo Liang, Jing Ma, Huan Wang, and Yu Yin. Segment then splat: Unified 3d open-vocabulary segmentation via gaussian splatting. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2025.

[21] Ashkan Mirzaei, Tristan Aumentado-Armstrong, Konstantinos G. Derpanis, Jonathan Kelly, Marcus A. Brubaker, Igor Gilitschenski, and Alex Levinshtein. SPIn-NeRF: Multiview segmentation and perceptual inpainting with neural radiance fields. In *CVPR*, 2023.

[22] Minghan Qin, Wanhua Li, Jiawei Zhou, Haoqian Wang, and Hanspeter Pfister. LangSplat: 3D language Gaussian splatting. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2024.

[23] 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 *ACM International Conference on Multimedia (MM)*, 2024.

[24] Jin-Chuan Shi, Miao Wang, Hao-Bin Duan, and Shao-Hua Guan. Language embedded 3D Gaussians for open-vocabulary scene understanding. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2024.

[25] Yawar Siddiqui, Lorenzo Porzi, Samuel Rota Bulò, Norman Müller, Matthias Nießner, Angela Dai, and Peter Kontschieder. Panoptic lifting for 3D scene understanding with neural fields. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2023.

[26] Vadim Tschernezki, Iro Laina, Diane Larlus, and Andrea Vedaldi. Neural Feature Fusion Fields: 3D distillation of self-supervised 2D image representations. In *International Conference on 3D Vision (3DV)*, 2022.

[27] Chung-Ho Wu, Yang-Jung Chen, Ying-Huan Chen, Jie-Ying Lee, Bo-Hsu Ke, Chun-Wei Tuan Mu, Yi-Chuan Huang, Chin-Yang Lin, Min-Hung Chen, Yen-Yu Lin, and Yu-Lun Liu. Aurafusion360: Augmented unseen region alignment for reference-based 360deg unbounded scene inpainting. In *Proceedings of the Computer Vision and Pattern Recognition Conference (CVPR)*, pages 16366–16376, June 2025.

[28] Yanmin Wu, Jiarui Meng, Haijie Li, Chenming Wu, Yahao Shi, Xinhua Cheng, Chen Zhao, Haocheng Feng, Errui Ding, Jingdong Wang, and Jian Zhang. OpenGaussian: Towards point-level 3D Gaussian-based open vocabulary understanding. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2024.

[29] Mingqiao Ye, Martin Danelljan, Fisher Yu, and Lei Ke. Gaussian grouping: Segment and edit anything in 3D scenes. In *European Conference on Computer Vision (ECCV)*, 2024.

[30] Haiyang Ying, Yixuan Yin, Jinzhi Zhang, Fan Wang, Tao Yu, Ruqi Huang, and Lu Fang. OmniSeg3D: Omniversal 3D segmentation via hierarchical contrastive learning. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2024.

[31] Shuaifeng Zhi, Tristan Laidlow, Stefan Leutenegger, and Andrew J. Davison. In-place scene labelling and understanding with implicit scene representation. In *IEEE/CVF International Conference on Computer Vision (ICCV)*, 2021.

[32] 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 *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2024.

[33] Ruijie Zhu, Mulin Yu, Linning Xu, Lihan Jiang, Yixuan Li, Tianzhu Zhang, Jiangmiao Pang, and Bo Dai. ObjectGS: Object-aware scene reconstruction and scene understanding via Gaussian splatting. In *IEEE/CVF International Conference on Computer Vision (ICCV)*, 2025.

[34] Xingxing Zuo, Pouya Samangouei, Yunwen Zhou, Yan Di, and Mingyang Li. FMGS: Foundation model embedded 3D Gaussian splatting for holistic 3D scene understanding. *International Journal of Computer Vision*, 2024.

A Appendix

A.1 Implementation Details

We implement VODA in PyTorch on top of the official codebase of each baseline and run the main experiments on a single NVIDIA RTX A6000 GPU. VODA changes only feature learning: data preparation, scene reconstruction, the feature representation, and the inference-time Gaussian selection procedure remain unchanged. At an interval of T_{int} iterations, we run the depth-aware rasterizer and add $\mathcal{L}_{\text{direct}}$ and $\mathcal{L}_{\text{push}}$ to the native baseline objective.

Baseline integration. For OpenGaussian [28] and LaGa [6], we enable VODA when instance-feature learning begins, after scene reconstruction, with $T_{\text{int}} = 5$. For OpenGaussian, we map the Gaussian features and rendered targets to $[0, 1]$ before applying the squared-ℓ_2 direct loss. Subsequent codebook and language-association stages remain unchanged.

LaGa learns a 32-dimensional feature with a cosine-based contrastive objective. We therefore replace the squared-ℓ_2 distance in Eq. 6 with cosine distance. Here, $\hat{\mathbf{F}}_i$ is the $\alpha_i(p)$-weighted aggregate of pixel-level rendered feature targets for g_i, as defined in Eq. 5:

$$\mathcal{L}_{\text{direct}}^{\text{LaGa}} \;=\; \frac{1}{|\mathcal{G}_{\text{sel}}|} \sum_{g_i \in \mathcal{G}_{\text{sel}}} \Big(1 \;-\; \frac{\hat{\mathbf{F}}_i^{\top}\, \mathbf{f}_i}{\|\hat{\mathbf{F}}_i\|_2 \, \|\mathbf{f}_i\|_2 + \epsilon}\Big). \tag{10}$$

Gaussian Grouping [29] and ILGS [10] learn identity features jointly with scene reconstruction. For the reported experiments, we retain their standard training and append a $5{,}000$-iteration feature fine-tuning stage with $T_{\text{int}} = 1$. Geometry, appearance, and the identity classifier are frozen, while the existing Gaussian identity features are optimized with the original loss and the two VODA losses. Gaussian Grouping retains its original 3D consistency term. For ILGS, the CLIP-aligned semantic feature is also frozen, and VODA is applied only to the identity feature used for instance grouping.

Hyperparameters. Across all four baselines, we use $\gamma = 15$ and clip the adaptive depth threshold to $[\tau_{\min}, \tau_{\max}] = [0.05, 1.0]$. We set $\lambda_{\text{direct}} = 0.5$, $\lambda_{\text{push}} = 0.1$, and the push margin to $m = 0.3$. These values are shared without per-baseline tuning, while only the integration schedule differs.

A.2 Removal-3D Benchmark and RDE Computation

Dataset. Removal-3D contains three synthetic indoor scenes—Table, Kitchen, and Room—constructed in Blender using BlenderKit assets. For every target, the benchmark provides the full scene and a counterpart with that object removed, along with object masks, depth maps, and camera parameters estimated with COLMAP.

Depth rendering and alignment. For target object o in view f, let $D_{\text{gt}}^{\text{full}}(u)$ be the benchmark depth at pixel u in the full scene, and let $D_{\text{gt}}^{\text{rem},(o)}(u)$ be the depth in the paired scene with o removed. Their signed difference is

$$\Delta D_{\text{gt}}^{(o)}(u) \;=\; D_{\text{gt}}^{\text{full}}(u) - D_{\text{gt}}^{\text{rem},(o)}(u). \tag{11}$$

We render predicted depth before and after removing the selected Gaussian set using the standard alpha-compositing weights:

$$D_{\text{pred}}(u) \;=\; \sum_{i \in \mathcal{G}(u)} T_i(u)\, \alpha_i(u)\, z_i(u), \tag{12}$$

where $z_i(u)$ is the camera-space depth of g_i. The predicted depth is not calibrated to the benchmark depth, so we fit a global scale and offset using the full-scene prediction and its ground-truth depth:

$$(s^*,\, t^*) \;=\; \arg\min_{s,t} \sum_{u \in \mathcal{V}} \big(s\, D_{\text{pred}}^{\text{full}}(u) + t - D_{\text{gt}}^{\text{full}}(u)\big)^2, \tag{13}$$

where $\mathcal{V} = \{u : D_{\text{pred}}^{\text{full}}(u) > 0 \,\wedge\, D_{\text{gt}}^{\text{full}}(u) > 0\}$. We apply the same transform to the predicted removal depth. Subtracting the aligned depths cancels the offset, giving $\Delta D_{\text{pred}}^{(o)}(u) = s^*(D_{\text{pred}}^{\text{full}}(u) - D_{\text{pred}}^{\text{rem},(o)}(u))$.

Removal Depth Error. We compute the RDE over the ground-truth mask m_o of target object o, using the pixels $\Omega_o = \{u : m_o(u) = 1\}$. Before computing the mean absolute error, we clip the signed predicted depth change to the symmetric interval defined by the reference magnitude:

$$\widetilde{\Delta D}^{(o)}_{\mathrm{pred}}(u) = \mathrm{clip}\Big(\Delta D^{(o)}_{\mathrm{pred}}(u), -|\Delta D^{(o)}_{\mathrm{gt}}(u)|, |\Delta D^{(o)}_{\mathrm{gt}}(u)|\Big). \tag{14}$$

The reported RDE is

$$\mathrm{RDE}(o, f) = \frac{1}{|\Omega_o|} \sum_{u \in \Omega_o} \left| \widetilde{\Delta D}^{(o)}_{\mathrm{pred}}(u) - \Delta D^{(o)}_{\mathrm{gt}}(u) \right|. \tag{15}$$

Evaluation and aggregation. For each scene, we evaluate the eight predefined test views (`frame_00`, `frame_20`, ..., `frame_140`). We average the per-view RDE over views for each object, macro-average over objects to obtain each scene's mRDE, and report the mean across scenes:

$$\text{per-object RDE} = \tfrac{1}{|\mathcal{F}_o|} \sum_{f \in \mathcal{F}_o} \mathrm{RDE}(o, f), \tag{16}$$

$$\text{per-scene mRDE} = \tfrac{1}{|\mathcal{O}_s|} \sum_{o \in \mathcal{O}_s} \text{per-object RDE}(o), \tag{17}$$

$$\text{total mRDE} = \tfrac{1}{|\mathcal{S}|} \sum_{s \in \mathcal{S}} \text{per-scene mRDE}(s). \tag{18}$$

We also report rendered-mask mIoU on the same views by projecting $\hat{\mathcal{G}}_o$ and comparing it with the ground-truth mask m_o.

A.3 Additional Baseline Comparison and Positioning

Reconstruction-time correspondence: Segment then Splat. Segment then Splat (STS) [20] and VODA both seek object-consistent Gaussian representations, but introduce object identity at different stages. STS partitions Gaussians into object-specific sets using pre-obtained masks and optimizes the scene under this fixed correspondence. VODA instead operates on an already-reconstructed scene: it adds direct object-level supervision to the baseline's features while leaving geometry, opacity, and appearance unchanged. STS therefore establishes object-specific Gaussian sets during reconstruction, whereas VODA improves feature-based object assignment after reconstruction.

Volumetric segmentation and reconstruction quality. We evaluate the public STS implementation under the same Removal-3D protocol (Table 7). STS achieves higher selection mIoU than Gaussian Grouping (38.37 vs. 26.23), but has a higher mRDE than either unmodified baseline (62.10 vs. 39.31 and 50.05). In contrast, VODA improves both metrics for each baseline without retraining its geometry. Table 8 also reports full-scene reconstruction quality averaged over the three scenes. VODA leaves these metrics unchanged because it modifies only object features. In this setting, STS has lower PSNR and SSIM and higher LPIPS than the two baselines. These results show that reconstruction-time correspondence alone does not ensure volumetrically complete object selection. However, STS and the feature-learning baselines use different reconstructed scenes, so this comparison does not isolate object assignment from reconstruction quality. The baseline–VODA comparisons keep the reconstructed scene fixed and isolate the effect of feature supervision.

Complementary scope and trade-offs. Fixed mask-based correspondence can propagate errors from partial views or ambiguous boundaries into both object assignment and geometry optimization. This dependence makes reliable masks and cross-view tracking important for reconstruction-time methods. VODA decouples feature supervision from reconstruction and can be applied to pre-trained scenes, but cannot repair errors in the geometry it inherits. STS is attractive when a reliable initial object decomposition is available, while VODA addresses incomplete selection in existing rendering-based feature-learning pipelines. We therefore view the approaches as complementary rather than claiming that either paradigm universally dominates the other.

Table 7: Object selection on Removal-3D.

Method	mIoU ↑	mRDE ↓
OpenGaussian	52.87	39.31
OpenGaussian + VODA	**60.90**	**27.34**
Gaussian Grouping	26.23	50.05
Gaussian Grouping + VODA	30.18	38.12
Segment then Splat	38.37	62.10

Table 8: RGB reconstruction on Removal-3D.

Method	PSNR ↑	SSIM ↑	LPIPS ↓
OpenGaussian	33.4	0.925	0.105
Gaussian Grouping	34.7	0.954	0.069
Segment then Splat	28.2	0.884	0.237

A.4 Diagnostic Analyses

Scene-level contribution statistics. The main-paper analysis (Fig. 7) shows that Gaussians recovered by VODA have lower maximum rendering contributions on Removal-3D. We extend this analysis across six scenes from LERF and Removal-3D using accumulated-contribution statistics.

For each Gaussian, we compute its accumulated contribution by summing $w_i(p) = T_i(p)\alpha_i(p)$ over training pixels and views. We also record its maximum per-pixel contribution. We compare baseline-retrieved Gaussians (*selected*) with those missed by the baseline but retrieved after adding VODA (*missed*). Both sets are measured in the same reconstructed scene. Because the baseline and VODA checkpoints share geometry and opacity, each Gaussian has identical rendering contributions across the two checkpoints. Across Removal-3D scenes, VODA excludes 1.9–6.8% of the Gaussians selected by the baseline. Excluding these Gaussians changes the cutoff fractions by at most 0.3 percentage points.

Table 9 shows that, across all six scenes, the missed set has lower median accumulated contribution and higher fractions of low-maximum and zero-contribution Gaussians than the selected set.

Table 9: Rendering-contribution statistics for LaGa. Selected denotes baseline-retrieved Gaussians, and Missed denotes those recovered only with VODA. The median is computed over accumulated training-view contributions. The final columns give fractions below the maximum-contribution cutoff and with zero contribution in every view.

Dataset	Scene	Set	Median $\sum w$	$\max w < 1/255$ (%)	Zero (%)
LERF	Figurines	Selected	119.87	25.1	10.6
		Missed	0.23	59.3	18.7
	Ramen	Selected	29.70	15.1	6.4
		Missed	9.75	21.5	10.9
	Teatime	Selected	57.66	7.5	3.7
		Missed	15.48	18.6	8.5
Removal-3D	Kitchen	Selected	20.33	6.9	4.4
		Missed	4.95	17.1	10.3
	Room	Selected	26.50	10.5	4.7
		Missed	7.30	26.4	12.8
	Table	Selected	8.71	23.3	6.6
		Missed	1.72	36.5	14.3

Pixel-level push. The pixel-level push sharpens boundaries between adjacent instances that depth clustering alone may not separate. Figure 8 compares the rendered masks with and without this loss.

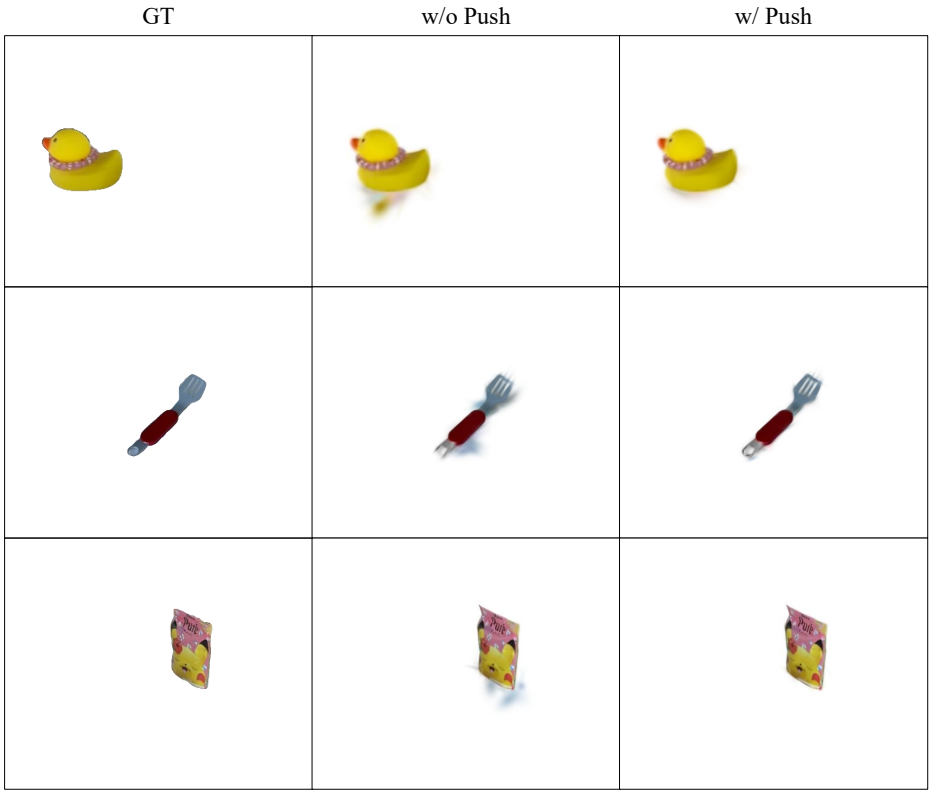

Figure 8: Effect of the pixel-level push loss. Ground truth (left), VODA without push (middle), and VODA with push (right). The push loss sharpens adjacent-instance boundaries.

A.5 Robustness and Efficiency

Depth-threshold sensitivity. We vary γ and the threshold clamps using Gaussian Grouping on LERF (Tables 10 and 11). Across the sweeps, mIoU spans 0.43 and 1.03 percentage points, respectively. The default $\gamma = 15$ is near-best. Raising or removing $\tau_{\max}$ lowers mIoU, while changes to $\tau_{\min}$ have smaller effects in the tested range. Dashes indicate omitted clamp bounds.

Table 10: Split sensitivity on LERF.

γ	mIoU ↑
5	43.40
10	43.58
15 (default)	43.73
20	43.83
25	43.51
30	43.70

Table 11: Threshold clamps on LERF.

$\tau_{\min}$	$\tau_{\max}$	mIoU ↑
0.05	1.0	43.73
0.025	1.0	44.13
0.075	1.0	43.65
0.05	0.5	43.97
0.05	1.5	43.19
–	1.0	44.13
0.05	–	43.10
–	–	43.17

Injection timing. Table 12 tests whether VODA can be optimized jointly with reconstruction. Enabling VODA from the start of training improves both mIoU and mRDE over the baseline (33.1 vs. 26.4 and 32.2 vs. 46.4), showing that VODA need not wait for geometry training to finish. Enabling it at 10K or 20K iterations also outperforms waiting until geometry training ends at 30K on both metrics. The 10K and 20K settings reach mIoU values of 32.1 and 32.2, compared with 31.1 at 30K, and mRDE values of 24.1 and 25.0, compared with 30.2 at 30K. The best timing depends on the

metric. Starting at 0 gives the highest mIoU, while starting at 10K gives the lowest mRDE in this sweep. The main experiments use post-training feature fine-tuning to keep geometry fixed and isolate the effect of feature supervision. These values come from a separate diagnostic run and differ slightly from the reported results.

Table 12: Sensitivity to the VODA injection point for Gaussian Grouping.

	Baseline	$t = 0$	$t = 10$K	$t = 20$K	$t = 30$K
mIoU ↑	26.4	33.1	32.1	32.2	31.1
mRDE ↓	46.4	32.2	24.1	25.0	30.2

Interior versus boundary RDE. We split the evaluation mask into an interior and a boundary to test whether VODA improves removal throughout the object or mainly near its silhouette. We define the interior by eroding the ground-truth mask by five pixels. It retains 95–97% of the object mask, and the removed border forms the boundary region. Table 13 shows comparable mRDE reductions in both regions for both baselines, indicating that the gains are not confined to boundary alignment.

Table 13: Interior and boundary mRDE on Removal-3D. The interior is obtained by eroding the object mask by five pixels; Δ denotes the reduction in mRDE relative to the corresponding baseline.

Method	Interior ↓	Boundary ↓	Δ Interior	Δ Boundary
OpenGaussian	39.22	39.09	–	–
+ VODA (−push)	31.46	32.42	7.8	6.7
+ VODA	**26.81**	**27.12**	12.4	12.0
Gaussian Grouping	49.80	46.82	–	–
+ VODA (−push)	39.23	**34.84**	10.6	12.0
+ VODA	**39.13**	34.96	10.7	11.9

Computational overhead. Table 14 reports timings over 1,000 iterations on LERF-figurines using an RTX A5000, with one baseline from each integration schedule. Most of the added cost comes from the no-gradient depth-aware rasterizer forward pass. Target aggregation and the additional backward pass are comparatively small. The per-iteration overhead applies only while VODA is active. It adds approximately 10 minutes to the OpenGaussian pipeline and 20 minutes to the Gaussian Grouping pipeline. VODA adds no inference-time cost because the baseline selection procedure is unchanged.

Table 14: Training overhead on LERF-figurines (RTX A5000). Per-iteration timings are averaged over 1,000 iterations. Full-pipeline values give the added training time.

	OpenGaussian ($T_{\text{int}} = 5$)	Gaussian Grouping ($T_{\text{int}} = 1$)
Baseline per iteration	124.1 ms	75.7 ms
+ VODA per iteration	183.3 ms (+47.7%)	302.3 ms (+299%)
Depth-aware rasterizer	55.8 ms	219.9 ms
Targets, losses, backward	∼3 ms	∼7 ms
Added full-pipeline time	∼10 min	∼20 min

A.6 Run-to-Run Variability

We repeat VODA training three times on each applicable benchmark while keeping baseline results fixed. Tables 15–17 report mean $\pm$ sample standard deviation ($n = 3$). The main tables report the first run.

Table 15: Run-to-run variability on LERF (mIoU ↑). Mean $\pm\sigma$ is computed over three VODA runs with fixed baseline results.

Method	Baseline	+ VODA (Mean $\pm\sigma$)
OpenGaussian	37.12	42.29 ± 0.80
LaGa	56.80	60.17 ± 0.78
Gaussian Grouping	22.75	44.07 ± 0.31
ILGS	30.00	49.27 ± 0.52

Table 16: Run-to-run variability on ScanNet (mIoU ↑). Mean $\pm\sigma$ is computed over three VODA runs for each class split.

Method	Classes	Baseline	+ VODA (Mean $\pm\sigma$)
OpenGaussian	19	26.20	28.32 ± 1.12
	15	28.00	28.75 ± 1.00
	10	34.57	35.93 ± 0.99
LaGa	19	32.20	34.51 ± 1.25
	15	37.19	38.26 ± 1.06
	10	43.06	45.51 ± 1.24

Table 17: Run-to-run variability on Removal-3D. Mean $\pm\sigma$ is computed over three VODA runs with fixed baseline results.

Method	Metric	Baseline	+ VODA (Mean $\pm\sigma$)
OpenGaussian	mIoU ↑	52.87	58.16 ± 2.62
	mRDE ↓	39.31	30.69 ± 2.92
LaGa	mIoU ↑	75.93	77.71 ± 1.53
	mRDE ↓	36.93	22.08 ± 0.45
Gaussian Grouping	mIoU ↑	26.23	30.23 ± 0.68
	mRDE ↓	50.05	36.74 ± 2.50
ILGS	mIoU ↑	31.66	37.62 ± 0.99
	mRDE ↓	49.98	36.59 ± 1.24

A.7 Real-World Quantitative and Perceptual Evaluation

Paired real-world captures. We evaluate object removal on SPIn-NeRF [21] (10 scenes) and 360-USID [27] (7 scenes), which provide captures of scenes both with and without a target object. For each scene, we reconstruct the with-object capture, retrieve the target Gaussian set with each baseline or its VODA-enhanced counterpart, remove the selected Gaussians, and render the result. We evaluate OpenGaussian and Gaussian Grouping, using object-mask IoU on annotated views to assess selection. For a real-world proxy of RDE, we reconstruct the with-object and without-object captures independently using vanilla 3DGS and use their rendered depth difference as the reference removal signature. Unlike Removal-3D, this reference is estimated from reconstructions rather than ground-truth geometry. We also compare removal renderings with the without-object reference images using PSNR, SSIM, and LPIPS.

VODA improves object-mask IoU in all four dataset–baseline settings. However, the proxy RDE changes are small, with a slight regression for OpenGaussian on 360-USID, and the image-space changes are mixed. These datasets mainly contain single-object, low-occlusion captures, in which baseline selections already recover most of the depth change. Moreover, image-space metrics conflate removal completeness with the quality of newly exposed regions, which were not observed during reconstruction. Thus, these evaluations support transfer to real reconstructions but do not directly establish volumetric completeness under heavy occlusion.

Table 18: Real-world object removal on paired-capture datasets. Each entry reports baseline $\rightarrow$ +VODA. Object-mask IoU measures selection accuracy, while proxy RDE uses a depth-change reference derived from separate 3DGS reconstructions of the paired captures.

Dataset	Method	Mask IoU ↑	Proxy RDE ↓
SPIn-NeRF (10)	OpenGaussian	$0.339 \rightarrow 0.493$	$9.22 \rightarrow 9.20$
	Gaussian Grouping	$0.675 \rightarrow 0.728$	$9.82 \rightarrow 9.78$
360-USID (7)	OpenGaussian	$0.195 \rightarrow 0.207$	$0.601 \rightarrow 0.609$
	Gaussian Grouping	$0.347 \rightarrow 0.553$	$3.034 \rightarrow 2.911$

Table 19: Image-space quality of real-world object removal, compared with the without-object reference captures. Each entry is baseline $\rightarrow$ +VODA.

Dataset	Method	PSNR ↑	SSIM ↑	LPIPS ↓
SPIn-NeRF	OpenGaussian	$18.70 \rightarrow 18.29$	$0.667 \rightarrow 0.666$	$0.329 \rightarrow 0.331$
	Gaussian Grouping	$19.79 \rightarrow 19.82$	$0.711 \rightarrow 0.711$	$0.273 \rightarrow 0.272$
360-USID	OpenGaussian	$17.79 \rightarrow 17.88$	$0.565 \rightarrow 0.571$	$0.316 \rightarrow 0.313$
	Gaussian Grouping	$18.56 \rightarrow 19.08$	$0.563 \rightarrow 0.569$	$0.262 \rightarrow 0.252$

VLM-based perceptual evaluation. We additionally evaluate removal renderings from real LERF scenes using a VLM judge (GPT-5.6). Baseline and +VODA results are scored independently on a five-point quality scale, with method identities hidden and left–right placement balanced. For each paired comparison, a win indicates a higher score for that method; equal scores are counted as ties. Table 20 reports the comparison counts.

Table 20: VLM preferences for real-scene (LERF) removal quality. n counts evaluated cases; ties are cases with equal scores.

Method	n	Baseline wins	+VODA wins	Ties
Overall	80	12	41	27
OpenGaussian	57	12	21	24
Gaussian Grouping	23	0	20	3

The VLM judge favors VODA in the number of non-tied comparisons, particularly for Gaussian Grouping. These judgments provide complementary perceptual evidence on real scenes without

requiring paired removal captures; they do not replace the ground-truth depth evaluation on Removal-3D.

A.8 Qualitative Results and Failure Modes

Real-world results on LERF. To complement the synthetic Removal-3D benchmark, we show qualitative object queries on real LERF scenes in Fig. 9. The examples illustrate object selection and scene rendering after Gaussian removal in captured environments.

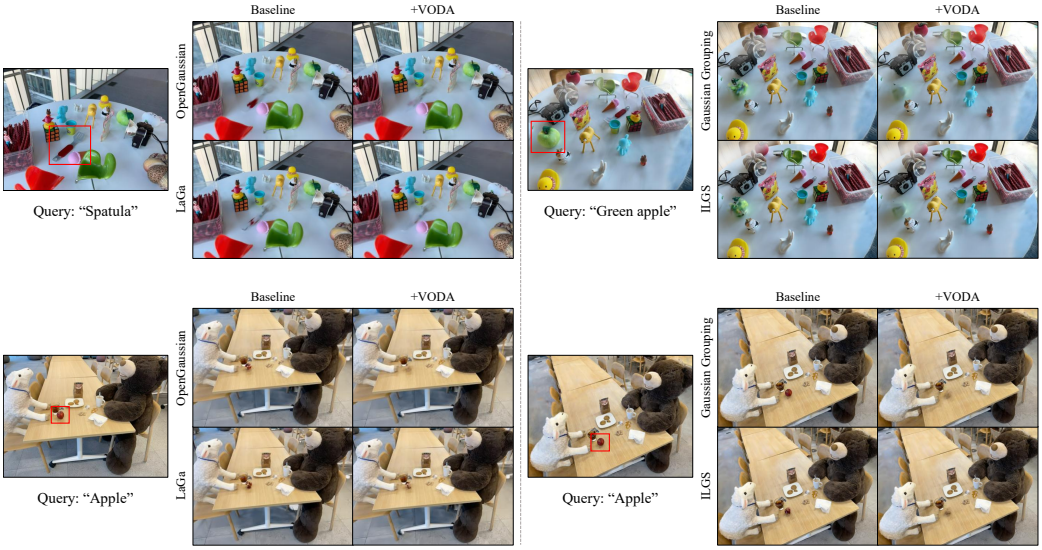

Figure 9: Qualitative object selection and Gaussian removal on real LERF scenes. Red boxes mark the queried objects, and the accompanying renderings show the corresponding scene views.

Failure cases. Failure examples highlight two recurring limitations. Depth clustering can under-split contacting objects with similar depths, while sparse object interiors can leave residual Gaussians after removal.



Figure 10: Failure case on a real LERF scene. The queried teddy bear is outlined in red in the input view, and the renderings after Gaussian removal show substantial artifacts around the target region.

NeurIPS Paper Checklist

1. **Claims**

   Question: Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope?

   Answer: [Yes]

   Justification: The main claims made in the abstract and introduction accurately reflect the paper's contributions and scope.

   Guidelines:

   - The answer [N/A] means that the abstract and introduction do not include the claims made in the paper.
   - The abstract and/or introduction should clearly state the claims made, including the contributions made in the paper and important assumptions and limitations. A [No] or [N/A] answer to this question will not be perceived well by the reviewers.
   - The claims made should match theoretical and experimental results, and reflect how much the results can be expected to generalize to other settings.
   - It is fine to include aspirational goals as motivation as long as it is clear that these goals are not attained by the paper.

2. **Limitations**

   Question: Does the paper discuss the limitations of the work performed by the authors?

   Answer: [Yes]

   Justification: In the end part of our main paper, we discuss the limitations of this paper.

   Guidelines:

   - The answer [N/A] means that the paper has no limitation while the answer [No] means that the paper has limitations, but those are not discussed in the paper.
   - The authors are encouraged to create a separate "Limitations" section in their paper.
   - The paper should point out any strong assumptions and how robust the results are to violations of these assumptions (e.g., independence assumptions, noiseless settings, model well-specification, asymptotic approximations only holding locally). The authors should reflect on how these assumptions might be violated in practice and what the implications would be.
   - The authors should reflect on the scope of the claims made, e.g., if the approach was only tested on a few datasets or with a few runs. In general, empirical results often depend on implicit assumptions, which should be articulated.
   - The authors should reflect on the factors that influence the performance of the approach. For example, a facial recognition algorithm may perform poorly when image resolution is low or images are taken in low lighting. Or a speech-to-text system might not be used reliably to provide closed captions for online lectures because it fails to handle technical jargon.
   - The authors should discuss the computational efficiency of the proposed algorithms and how they scale with dataset size.
   - If applicable, the authors should discuss possible limitations of their approach to address problems of privacy and fairness.
   - While the authors might fear that complete honesty about limitations might be used by reviewers as grounds for rejection, a worse outcome might be that reviewers discover limitations that aren't acknowledged in the paper. The authors should use their best judgment and recognize that individual actions in favor of transparency play an important role in developing norms that preserve the integrity of the community. Reviewers will be specifically instructed to not penalize honesty concerning limitations.

3. **Theory assumptions and proofs**

   Question: For each theoretical result, does the paper provide the full set of assumptions and a complete (and correct) proof?

   Answer: [N/A]

Justification: This paper does not present formal theoretical results requiring proofs.

Guidelines:

- The answer [N/A] means that the paper does not include theoretical results.
- All the theorems, formulas, and proofs in the paper should be numbered and cross-referenced.
- All assumptions should be clearly stated or referenced in the statement of any theorems.
- The proofs can either appear in the main paper or the supplemental material, but if they appear in the supplemental material, the authors are encouraged to provide a short proof sketch to provide intuition.
- Inversely, any informal proof provided in the core of the paper should be complemented by formal proofs provided in appendix or supplemental material.
- Theorems and Lemmas that the proof relies upon should be properly referenced.

4. **Experimental result reproducibility**

   Question: Does the paper fully disclose all the information needed to reproduce the main experimental results of the paper to the extent that it affects the main claims and/or conclusions of the paper (regardless of whether the code and data are provided or not)?

   Answer: [Yes]

   Justification: We will provide open-source codes.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - If the paper includes experiments, a [No] answer to this question will not be perceived well by the reviewers: Making the paper reproducible is important, regardless of whether the code and data are provided or not.
   - If the contribution is a dataset and/or model, the authors should describe the steps taken to make their results reproducible or verifiable.
   - Depending on the contribution, reproducibility can be accomplished in various ways. For example, if the contribution is a novel architecture, describing the architecture fully might suffice, or if the contribution is a specific model and empirical evaluation, it may be necessary to either make it possible for others to replicate the model with the same dataset, or provide access to the model. In general. releasing code and data is often one good way to accomplish this, but reproducibility can also be provided via detailed instructions for how to replicate the results, access to a hosted model (e.g., in the case of a large language model), releasing of a model checkpoint, or other means that are appropriate to the research performed.
   - While NeurIPS does not require releasing code, the conference does require all submissions to provide some reasonable avenue for reproducibility, which may depend on the nature of the contribution. For example
     (a) If the contribution is primarily a new algorithm, the paper should make it clear how to reproduce that algorithm.
     (b) If the contribution is primarily a new model architecture, the paper should describe the architecture clearly and fully.
     (c) If the contribution is a new model (e.g., a large language model), then there should either be a way to access this model for reproducing the results or a way to reproduce the model (e.g., with an open-source dataset or instructions for how to construct the dataset).
     (d) We recognize that reproducibility may be tricky in some cases, in which case authors are welcome to describe the particular way they provide for reproducibility. In the case of closed-source models, it may be that access to the model is limited in some way (e.g., to registered users), but it should be possible for other researchers to have some path to reproducing or verifying the results.

5. **Open access to data and code**

   Question: Does the paper provide open access to the data and code, with sufficient instructions to faithfully reproduce the main experimental results, as described in supplemental material?

Answer: [N/A]

Justification: We will provide it later.

Guidelines:

- The answer [N/A] means that paper does not include experiments requiring code.
- Please see the NeurIPS code and data submission guidelines (`https://neurips.cc/public/guides/CodeSubmissionPolicy`) for more details.
- While we encourage the release of code and data, we understand that this might not be possible, so [No] is an acceptable answer. Papers cannot be rejected simply for not including code, unless this is central to the contribution (e.g., for a new open-source benchmark).
- The instructions should contain the exact command and environment needed to run to reproduce the results. See the NeurIPS code and data submission guidelines (`https://neurips.cc/public/guides/CodeSubmissionPolicy`) for more details.
- The authors should provide instructions on data access and preparation, including how to access the raw data, preprocessed data, intermediate data, and generated data, etc.
- The authors should provide scripts to reproduce all experimental results for the new proposed method and baselines. If only a subset of experiments are reproducible, they should state which ones are omitted from the script and why.
- At submission time, to preserve anonymity, the authors should release anonymized versions (if applicable).
- Providing as much information as possible in supplemental material (appended to the paper) is recommended, but including URLs to data and code is permitted.

6. **Experimental setting/details**

   Question: Does the paper specify all the training and test details (e.g., data splits, hyperparameters, how they were chosen, type of optimizer) necessary to understand the results?

   Answer: [Yes]

   Justification: The details of training and testing are provided in A.1.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - The experimental setting should be presented in the core of the paper to a level of detail that is necessary to appreciate the results and make sense of them.
   - The full details can be provided either with the code, in appendix, or as supplemental material.

7. **Experiment statistical significance**

   Question: Does the paper report error bars suitably and correctly defined or other appropriate information about the statistical significance of the experiments?

   Answer: [Yes]

   Justification: We write them in Appendix.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - The authors should answer [Yes] if the results are accompanied by error bars, confidence intervals, or statistical significance tests, at least for the experiments that support the main claims of the paper.
   - The factors of variability that the error bars are capturing should be clearly stated (for example, train/test split, initialization, random drawing of some parameter, or overall run with given experimental conditions).
   - The method for calculating the error bars should be explained (closed form formula, call to a library function, bootstrap, etc.)
   - The assumptions made should be given (e.g., Normally distributed errors).
   - It should be clear whether the error bar is the standard deviation or the standard error of the mean.

- It is OK to report 1-sigma error bars, but one should state it. The authors should preferably report a 2-sigma error bar than state that they have a 96% CI, if the hypothesis of Normality of errors is not verified.
- For asymmetric distributions, the authors should be careful not to show in tables or figures symmetric error bars that would yield results that are out of range (e.g., negative error rates).
- If error bars are reported in tables or plots, the authors should explain in the text how they were calculated and reference the corresponding figures or tables in the text.

8. **Experiments compute resources**

   Question: For each experiment, does the paper provide sufficient information on the computer resources (type of compute workers, memory, time of execution) needed to reproduce the experiments?

   Answer: [Yes]

   Justification: We write them in Appendix.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - The paper should indicate the type of compute workers CPU or GPU, internal cluster, or cloud provider, including relevant memory and storage.
   - The paper should provide the amount of compute required for each of the individual experimental runs as well as estimate the total compute.
   - The paper should disclose whether the full research project required more compute than the experiments reported in the paper (e.g., preliminary or failed experiments that didn't make it into the paper).

9. **Code of ethics**

   Question: Does the research conducted in the paper conform, in every respect, with the NeurIPS Code of Ethics `https://neurips.cc/public/EthicsGuidelines`?

   Answer: [Yes]

   Justification: The research conducted in the paper conforms, in every respect, with the NeurIPS Code of Ethics.

   Guidelines:

   - The answer [N/A] means that the authors have not reviewed the NeurIPS Code of Ethics.
   - If the authors answer [No], they should explain the special circumstances that require a deviation from the Code of Ethics.
   - The authors should make sure to preserve anonymity (e.g., if there is a special consideration due to laws or regulations in their jurisdiction).

10. **Broader impacts**

    Question: Does the paper discuss both potential positive societal impacts and negative societal impacts of the work performed?

    Answer: [N/A]

    Justification: There is no societal impact of the work performed.

    Guidelines:

    - The answer [N/A] means that there is no societal impact of the work performed.
    - If the authors answer [N/A] or [No], they should explain why their work has no societal impact or why the paper does not address societal impact.
    - Examples of negative societal impacts include potential malicious or unintended uses (e.g., disinformation, generating fake profiles, surveillance), fairness considerations (e.g., deployment of technologies that could make decisions that unfairly impact specific groups), privacy considerations, and security considerations.

- The conference expects that many papers will be foundational research and not tied to particular applications, let alone deployments. However, if there is a direct path to any negative applications, the authors should point it out. For example, it is legitimate to point out that an improvement in the quality of generative models could be used to generate Deepfakes for disinformation. On the other hand, it is not needed to point out that a generic algorithm for optimizing neural networks could enable people to train models that generate Deepfakes faster.
- The authors should consider possible harms that could arise when the technology is being used as intended and functioning correctly, harms that could arise when the technology is being used as intended but gives incorrect results, and harms following from (intentional or unintentional) misuse of the technology.
- If there are negative societal impacts, the authors could also discuss possible mitigation strategies (e.g., gated release of models, providing defenses in addition to attacks, mechanisms for monitoring misuse, mechanisms to monitor how a system learns from feedback over time, improving the efficiency and accessibility of ML).

11. **Safeguards**

    Question: Does the paper describe safeguards that have been put in place for responsible release of data or models that have a high risk for misuse (e.g., pre-trained language models, image generators, or scraped datasets)?

    Answer: [N/A]

    Justification: Not related.

    Guidelines:

    - The answer [N/A] means that the paper poses no such risks.
    - Released models that have a high risk for misuse or dual-use should be released with necessary safeguards to allow for controlled use of the model, for example by requiring that users adhere to usage guidelines or restrictions to access the model or implementing safety filters.
    - Datasets that have been scraped from the Internet could pose safety risks. The authors should describe how they avoided releasing unsafe images.
    - We recognize that providing effective safeguards is challenging, and many papers do not require this, but we encourage authors to take this into account and make a best faith effort.

12. **Licenses for existing assets**

    Question: Are the creators or original owners of assets (e.g., code, data, models), used in the paper, properly credited and are the license and terms of use explicitly mentioned and properly respected?

    Answer: [Yes]

    Justification: The models and data used in this paper are open-sourced and authorized.

    Guidelines:

    - The answer [N/A] means that the paper does not use existing assets.
    - The authors should cite the original paper that produced the code package or dataset.
    - The authors should state which version of the asset is used and, if possible, include a URL.
    - The name of the license (e.g., CC-BY 4.0) should be included for each asset.
    - For scraped data from a particular source (e.g., website), the copyright and terms of service of that source should be provided.
    - If assets are released, the license, copyright information, and terms of use in the package should be provided. For popular datasets, `paperswithcode.com/datasets` has curated licenses for some datasets. Their licensing guide can help determine the license of a dataset.
    - For existing datasets that are re-packaged, both the original license and the license of the derived asset (if it has changed) should be provided.

- If this information is not available online, the authors are encouraged to reach out to the asset's creators.

13. **New assets**

    Question: Are new assets introduced in the paper well documented and is the documentation provided alongside the assets?

    Answer: [N/A]

    Justification: The paper does not release new assets.

    Guidelines:

    - The answer [N/A] means that the paper does not release new assets.
    - Researchers should communicate the details of the dataset/code/model as part of their submissions via structured templates. This includes details about training, license, limitations, etc.
    - The paper should discuss whether and how consent was obtained from people whose asset is used.
    - At submission time, remember to anonymize your assets (if applicable). You can either create an anonymized URL or include an anonymized zip file.

14. **Crowdsourcing and research with human subjects**

    Question: For crowdsourcing experiments and research with human subjects, does the paper include the full text of instructions given to participants and screenshots, if applicable, as well as details about compensation (if any)?

    Answer: [N/A]

    Justification: No human-participant evaluation is reported in the final manuscript.

    Guidelines:

    - The answer [N/A] means that the paper does not involve crowdsourcing nor research with human subjects.
    - Including this information in the supplemental material is fine, but if the main contribution of the paper involves human subjects, then as much detail as possible should be included in the main paper.
    - According to the NeurIPS Code of Ethics, workers involved in data collection, curation, or other labor should be paid at least the minimum wage in the country of the data collector.

15. **Institutional review board (IRB) approvals or equivalent for research with human subjects**

    Question: Does the paper describe potential risks incurred by study participants, whether such risks were disclosed to the subjects, and whether Institutional Review Board (IRB) approvals (or an equivalent approval/review based on the requirements of your country or institution) were obtained?

    Answer: [N/A]

    Justification: No human-participant evaluation is reported in the final manuscript.

    Guidelines:

    - The answer [N/A] means that the paper does not involve crowdsourcing nor research with human subjects.
    - Depending on the country in which research is conducted, IRB approval (or equivalent) may be required for any human subjects research. If you obtained IRB approval, you should clearly state this in the paper.
    - We recognize that the procedures for this may vary significantly between institutions and locations, and we expect authors to adhere to the NeurIPS Code of Ethics and the guidelines for their institution.
    - For initial submissions, do not include any information that would break anonymity (if applicable), such as the institution conducting the review.

16. **Declaration of LLM usage**

Question: Does the paper describe the usage of LLMs if it is an important, original, or non-standard component of the core methods in this research? Note that if the LLM is used only for writing, editing, or formatting purposes and does *not* impact the core methodology, scientific rigor, or originality of the research, declaration is not required.

Answer: [Yes]

Justification: To polish writing.

Guidelines:

- The answer [N/A] means that the core method development in this research does not involve LLMs as any important, original, or non-standard components.
- Please refer to our LLM policy in the NeurIPS handbook for what should or should not be described.