OV-FOAM: HOW 3D REPRESENTATIONS SHAPE SEMANTIC FEATURE LIFTING

Anonymous authors
Paper under double-blind review

ABSTRACT

Existing feature lifting approaches for open-vocabulary segmentation rely on simple backprojection to assign features to primitives. In the case of 3D Gaussian splatting reconstruction, the primitives overlap heavily, leading to suboptimal reconstructions, governed by many primitives per ray. Backprojection, however, reduces to a least-squares solvable problem, exactly when no two primitives share a ray. Motivated by this insight, we propose OV-Foam, which uses PowerFoam as the underlying representation. This representation uses a power-diagram to partition the scene, resulting in zero overlap between primitives and a substantial reduction in primitives per ray, which ultimately simplifies the lifting problem and allows for closer-to-optimal feature assignment. We show that the space partitioning primitives of OV-Foam reach near-optimal assignment, and our experiments demonstrate improved open-vocabulary segmentation over Gaussian baselines.

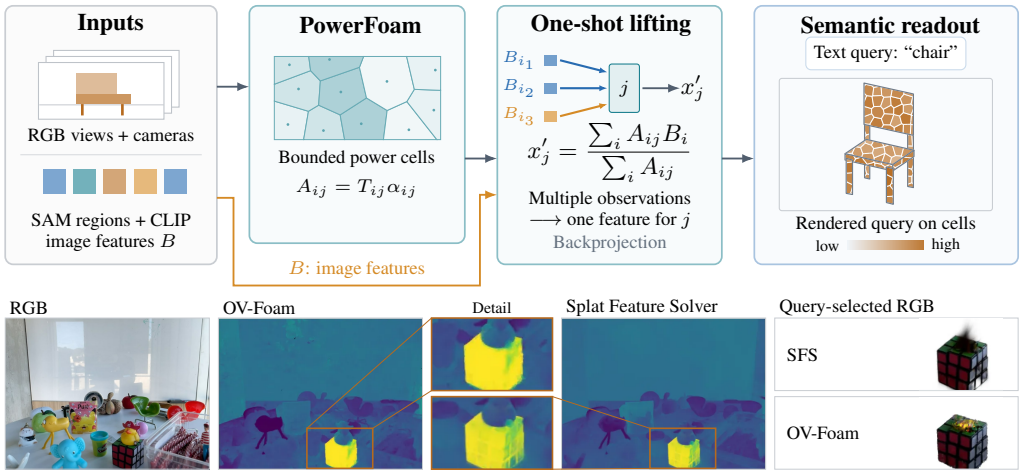


Figure 1: **Pipeline overview and qualitative example.** Starting from the input views, we create SAM masks, which are fed to CLIP to create image features B. The views also reconstruct our space-partitioning PowerFoam scene, of which we then use the rendering weights A for the back projection and assign each primitive a queryable feature x' that is compared to the text features.

1 INTRODUCTION

Open-vocabulary 3D segmentation assigns vision-language aligned semantic features, such as those produced by CLIP (Radford et al., 2021b), to reconstructed scenes, allowing objects and spatial regions to be identified using free-form language queries. This capability supports applications such as robotic manipulation (Gu et al., 2024), complex scene understanding (Kerr et al., 2023) and medical-imaging (Wang et al., 2025). Existing approaches include grouping-based methods, training-based methods, and heuristic-based methods. We focus on the latter family, which directly lifts features from 2D images onto a 3D representation.

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

The process of assigning these features to scene geometry is known as *feature lifting*. It associates features observed in rendered images with the primitives contributing to each pixel, using their rendering weights to aggregate information across views. This provides an efficient way to obtain semantic representations without training a 3D foundation model from scratch, but introduces a fundamental ambiguity: a 2D observation may be distributed across multiple primitives, while a primitive may collect observations from different objects. As a result, feature lifting can entangle semantic evidence before a classifier makes a prediction.

This ambiguity is particularly important near object boundaries. Repeating the lifting process across views can produce feature entanglement and degraded semantic boundaries, even when the underlying image features are accurate. Recent work has begun to formalize this problem by treating feature lifting as a linear inverse problem and quantifying the ambiguity introduced by multiple primitive contributions (Xiong et al., 2026).

Existing feature-lifting methods largely operate on Gaussian-based reconstructions, where overlap leads to multiple primitives contributing to the same ray. This raises a more fundamental question: to what extent is feature-lifting ambiguity a property of the lifting procedure itself, and to what extent is it induced by the underlying 3D representation?

In this work, we investigate feature lifting as a *representation-dependent* problem. We ask whether changing the geometric primitives used to represent the scene can reduce the mixing of semantic evidence, while keeping the image features and lifting procedure fixed. We consider PowerFoam (Govindarajan et al., 2026), which represents a scene using bounded power diagrams whose cells partition 3D space. This non-overlapping spatial support reduces the number of primitives that can contribute simultaneously to a ray, reducing semantic feature mixing.

We isolate this effect, by evaluating feature lifting independently of errors introduced by the upstream 2D semantic model. We replace predicted image semantics with ground-truth segmentation and text features, allowing us to measure the extent of semantic feature mixing caused by a feature lifting onto a chosen underlying 3D representation. Under this controlled setting, we compare Gaussian-based and PowerFoam reconstructions using the same lifting procedure and find that the reduced dispersion in PowerFoam leads to substantially more class-consistency in lifted features. This demonstrates that the choice of 3D primitive can itself impose a significant limitation on semantic feature lifting. Applied to real datasets, we observe that all our tested lifting methods achieve improved 3D open-vocabulary segmentation results using PowerFoam instead of 3DGS as underlying representation.

Our contributions are:

- **A controlled study of representation-dependent feature lifting.** We isolate the feature-lifting stage from upstream semantic errors and systematically compare 3D primitives under the same image features and lifting procedure.
- **An analysis of representation-induced feature dispersion.** We show that space-partitioning primitives can substantially reduce semantic feature mixing compared with overlapping Gaussian primitives.
- **An open-vocabulary 3D segmentation system based on PowerFoam.** We incorporate these insights into a PowerFoam-based semantic representation and demonstrate improved performance on open-vocabulary 3D segmentation benchmarks.

2 Related work

Existing approaches for lifting 2D features to 3D primitives differ primarily in how the correspondence between image observations and 3D primitives is established: through training-based, grouping-based, and heuristic-forward methods.

Early approaches fit continuous 3D semantic fields via scene-specific optimization. LERF, for example, distills CLIP features into a radiance field (Kerr et al., 2023; Radford et al., 2021b), while LangSplat trains a scene-specific autoencoder to represent lifted features in a latent space (Qin et al., 2024). OpenGaussian similarly learns instance-level features together with a codebook for point-level understanding (Wu et al., 2024). These methods can produce expressive semantic representa-

tions, but require computationally expensive per-scene optimization and may inherit inconsistencies from the underlying multi-view 2D predictions.

An alternative is replacing continuous optimization with discrete associations derived from 2D masks (e.g., SAM (Kirillov et al., 2023)). LAGA decomposes scenes into object entities by clustering SAM mask descriptors across views (Cen et al., 2025), while THGS constructs Gaussian adjacency graphs with contrastive edge reweighting for hierarchical superpoint partitioning (Dai et al., 2025). LUDVIG avoids training altogether by combining multi-view weighted feature averaging with graph diffusion across adjacent primitives (Marrie et al., 2025). Although these approaches eliminate dense feature optimization, they introduce multi-stage grouping heuristics and remain sensitive to primitive spatial overlap.

More recent methods avoid additional optimization by directly inverting the image-to-primitive aggregation used during rendering. Occam's LGS (Cheng et al., 2025), CosegGaussians (Dou et al., 2025), and DrSplat (Jun-Seong et al., 2025) independently use row-normalized splatting equations as zero-shot feature lifter.

Splat Feature Solver (SFS) (Xiong et al., 2026) formulates feature lifting as a sparse linear inverse problem and provides a closed-form row-sum estimator with a provable approximation bound. Specifically, SFS defines β as a measure of the dispersion of the optimal lifted feature along a viewing ray. In the case where each ray intersects only a single primitive, each row of the system matrix contains a single non-zero entry of value one, yielding this zero-dispersion regime. Standard 3D Gaussians, however, have substantial spatial overlap, causing multiple primitives to contribute to the same ray and increasing feature dispersion (Figure 2). SFS therefore introduces Tikhonov Guidance, which modifies the opacity weights during lifting to increase diagonal dominance and reduce this coupling. Thus, while SFS provides a principled formulation and efficient solver, its lifting accuracy remains tied to the overlap structure of the underlying Gaussian primitives.

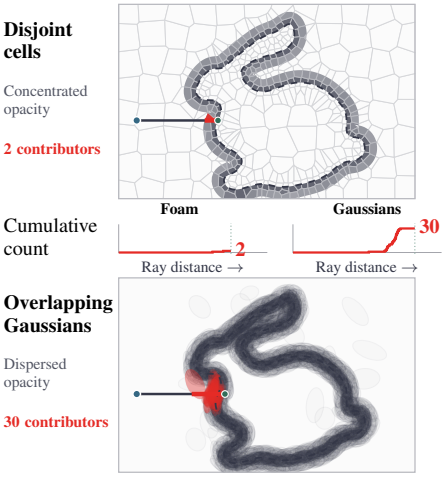


Figure 2: **Illustration of ray contributions.** Disjoint cells with concentrated opacity and overlapping Gaussians. Middle plots show cumulative contributor counts.

An alternative to mitigating primitive overlap through the lifting procedure is to eliminate overlap at the representation level. Space-partitioning representations divide the scene into disjoint spatial cells, so each 3D location is associated with a single primitive by construction. Radiant Foam (Govindarajan et al., 2025) represents a scene using non-overlapping volumetric cells based on an unbounded Voronoi construction. However, its ray-tracing procedure relies on explicit cell stepping rather than tile-based rasterization, making it difficult to evaluate under the same rendering pipeline as standard 3D Gaussian Splatting. PowerFoam (Govindarajan et al., 2026) instead generalizes foam representations to bounded power diagrams, defining finite spatial cells through seed sites and per-cell radii. These bounded cells support tile-based rasterization while preserving disjoint spatial boundaries, making PowerFoam a suitable representation for studying feature lifting under otherwise comparable rendering conditions.

Semantic Foam (Sharafeldin et al., 2026) jointly learns reconstruction and object-identity features supervised by segmentation masks. These features encode scene-specific identities rather than text-aligned semantics, so they cannot be directly compared with text embeddings for open-vocabulary queries. In contrast, we study feature transfer from fixed 2D observations onto an already trained reconstruction, allowing the effect of the underlying representation on lifting to be isolated from semantic optimization. Building on this setting, we introduce *OV-Foam*, an open-vocabulary 3D segmentation framework that lifts 2D vision-language features onto PowerFoam reconstructions.

3 METHOD

OV-Foam lifts image features onto a PowerFoam reconstruction. We first describe how the cells define rendering contributions, then use those contributions in the SFS feature estimator. We then examine how image observations are distributed across primitives and whether each primitive receives consistent features from one class or conflicting features from several classes.

3.1 POWERFOAM

PowerFoam represents a scene with a power diagram (Govindarajan et al., 2026). Each primitive j is associated with a center $\mathbf{p}_j$ and radius a_j. For a 3D point x, its power cell V_j is the region where primitive j has the smallest power distance among all primitives k. The rendering support C_j further restricts this cell to the primitive's bounding sphere:

$$\begin{aligned} V_j &= \{x : \|x - \mathbf{p}_j\|^2 - a_j^2 \le \|x - \mathbf{p}_k\|^2 - a_k^2 \text{ for all } k\}, \\ C_j &= V_j \cap \mathrm{Sphere}(\mathbf{p}_j, a_j). \end{aligned} \tag{1}$$

The power cells partition space, so their interiors do not overlap, although a ray can traverse several cells. Each primitive is bounded to C_j, so only the portion of its cell inside the bounding sphere can contribute to rendering.

Within each support C_j, a dipole defines an oriented boundary separating occupied space from empty space. The dipole provides a planar reference boundary, where occupied space satisfies $(\mathbf{x} - \mathbf{p}_j) \cdot \mathbf{n}_j < 0$, with $\mathbf{n}_j$ denoting the outward unit normal and $\mathbf{p}_j$ the primitive center. A displacement field can deform this boundary away from the reference plane. For a given ray i, PowerFoam determines the resulting intersections with both the support C_j and the occupied region. These intersections define the occupied ray segment and its length ℓ_{ij}. Density is constant within the occupied region and zero elsewhere. During reconstruction, primitive centers, radii, densities, dipoles, and enabled displacement parameters are optimized.

For a ray $r_i(t)$ parameterized by distance, volume rendering of a continuous 3D feature field $f : \mathbb{R}^3 \to \mathbb{R}^F$ between near and far limits t_n and t_f gives:

$$\mathcal{F}_i = \int_{t_n}^{t_f} T_i(t)\sigma(r_i(t))f(r_i(t))\,dt, \qquad T_i(t) = \exp\left(-\int_{t_n}^{t} \sigma(r_i(s))\,ds\right), \tag{2}$$

where $\mathcal{F}_i$ is the rendered feature for ray i, and $T_i(t)$ is the transmittance accumulated before distance t. PowerFoam discretizes $f(\mathbf{x})$ as a piecewise-constant field across primitives, where $f(\mathbf{x}) = x_j \in \mathbb{R}^F$ within the occupied region of primitive j and zero elsewhere. Because each primitive has constant density within its occupied region and carries a single feature x_j, the contribution of each occupied ray segment can be expressed through its length ℓ_{ij}. For primitive j, a segment of length ℓ_{ij} and density σ_j has opacity $\alpha_{ij} = 1 - \exp(-\sigma_j \ell_{ij})$. For the occupied segment of each primitive, the corresponding transmittance and opacity give the front-to-back weights:

$$A_{ij} = T_{ij}\alpha_{ij}, \qquad T_{ij} = \prod_{k \prec_i j} (1 - \alpha_{ik}), \qquad \mathcal{F}_i = \sum_j A_{ij} x_j, \quad s_i = \sum_j A_{ij} \le 1. \tag{3}$$

Here $k \prec_i j$ denotes a primitive encountered before j along ray i, and the background feature is zero. The total scene contribution can equivalently be written as $s_i = 1 - \prod_j (1 - \alpha_{ij})$, with the remaining $1 - s_i$ reaching the background. Thus, A_{ij} measures the contribution of primitive j to pixel i, after absorption by earlier primitives. A shorter occupied segment reduces α_{ij}, while opacity accumulated earlier along the ray reduces T_{ij}.

Gaussian splatting uses the same front-to-back compositing formulation, but obtains α_{ij} from the opacity and projected footprint of each Gaussian (Kerbl et al., 2023). PowerFoam instead integrates density through bounded occupied segments. Both renderers therefore provide weights A_{ij} for feature lifting, despite their different geometry. Spatial partitioning prevents overlapping occupied interiors, but contributions from different primitives can still accumulate along the same ray, depending on the reconstructed surface and density.

3.2 FEATURE LIFTING

Let $B \in \mathbb{R}^{R\times F}$ contain F feature values for each of the R rays, and let $X \in \mathbb{R}^{P\times F}$ contain the unknown features of P primitives. With the reconstruction fixed, the renderer defines a nonnegative contribution matrix $A \in \mathbb{R}^{R\times P}$, whose entries A_{ij} describe how much primitive j contributes to ray i. Rendering the primitive features X with these contributions produces the image features AX, which can be compared to the observed features B. SFS therefore formulates feature recovery as a least-squares inverse problem (Xiong et al., 2026):

$$\widehat{X} = \arg\min_X \mathcal{L}(X), \qquad \mathcal{L}(X) = \frac{1}{2}\|AX - B\|^2. \tag{4}$$

The minimum-norm least-squares is $A^\dagger B$, where $A^\dagger$ is the Moore–Penrose pseudoinverse. This solution minimizes the residual, but does not necessarily reproduce B exactly. We use $\widehat{X} = A^\dagger B$ to define dispersion in Section 3.3. The SFS one-shot approximation instead estimates each primitive feature by averaging the observations it receives:

$$x'_j = \frac{\sum_i A_{ij} B_i}{\sum_i A_{ij}}, \qquad X' = D^{-1}A^\top B, \qquad D = \mathrm{diag}(A^\top \mathbf{1}). \tag{5}$$

This expression applies to primitives with positive total contribution. Here $D_{jj} = \sum_i A_{ij}$ is that total and $\mathbf{1} \in \mathbb{R}^R$ is the all-ones vector. The average minimizes the separable objective

$$J(X) = \tfrac{1}{2}\sum_{i,j} A_{ij}\|x_j - B_i\|_2^2. \tag{6}$$

Unlike $\mathcal{L}$, J compares each primitive directly with each observation, rather than comparing their rendered mixture with that observation. SFS also gives a squared-weight variant after opacity adjustment in its Appendix B, while Equation 5 corresponds to its base estimator. Within the OV-Foam lifting framework, we instantiate this estimator on PowerFoam reconstructions by accumulating the numerator and denominator using PowerFoam's rendering weights. Real image observations use SAM regions encoded with CLIP (Kirillov et al., 2023; Radford et al., 2021a).

For this estimator, the relationship to the least-squares solution depends on the number and strength of contributors along each ray. When every ray has one fully absorbing contributor, Equation 5 is an exact least-squares solution. With multiple contributors, the unknown features are generally coupled, making the independent average an approximation. Even with a single contributor, however, weak opacity does not guarantee equivalence to least squares, since incomplete absorption changes the feature scale required to reproduce the observation. Equation 5 normalizes by $\sum_i A_{ij}$, whereas the normal equations normalize by $\sum_i A_{ij}^2$; these agree only when every $A_{ij} \in \{0, 1\}$.

3.3 DISPERSION AND CLASS PURITY

SFS introduces β as a measure of feature dispersion along each viewing ray, characterizing how closely the one-shot estimator approximates the least-squares solution. Under complete absorption, ray opacity $s_i = 1$. For each contributing primitive, define the feature discrepancy with respect to the observed pixel feature as $\Delta_{ij} = \|\widehat{x}_j - B_i\|_2$. Its weighted mean and relative dispersion are

$$\mu_i = \sum_j A_{ij}\Delta_{ij}, \qquad \beta_i = \frac{\sum_j A_{ij}(\Delta_{ij} - \mu_i)^2}{\mu_i^2}, \qquad \mu_i > 0. \tag{7}$$

The condition $s_i = 1$ turns A_{ij} into a probability distribution over contributors along ray i. Low β_i indicates that the primitives contributing to ray i share similar error distances.

In addition, we introduce purity P_j, which addresses the complementary question of which classes contribute evidence to a primitive. Given labeled image rays with classes y_i, let

$$\pi_{jc} = \frac{\sum_{i:y_i=c} A_{ij}}{\sum_{i:y_i \text{ is labeled}} A_{ij}}, \qquad P_j = \max_c \pi_{jc}. \tag{8}$$

Where π_{jc} is the share of the rendering *weight* that primitive j receives from class c, not the share of rays: one ray striking j with a large A_{ij} outweighs many that graze it, so the dominant class

by weight can differ from the dominant class by count (Figure 3). Let t_c be the unit CLIP text embedding of class c. With $B_i = t_{y_i}$ and the same labeled rays in both sums of Equation 5, the lifted feature becomes $x'j = \sum_c \pi j c t_c$, a convex combination of class embeddings in which P_j is the weight on the dominant class.

Inconsistent image observations arise from two distinct sources: geometric mixing, where rays sweep through multiple primitives across object boundaries, and upstream feature noise, where 2D models like CLIP produce different features across different viewpoints. These inconsistencies reveal a fundamental limitation of feature lifting: because each primitive j stores a single feature vector $x_j \in \mathbb{R}^F$, any primitive receiving conflicting observations across views is forced to compromise. Even the exact least-squares solution $\widehat{X} = A^\dagger B$ returns a blended feature vector (e.g., a centroid between competing class embeddings). While this centroid minimizes the quadratic rendering loss from Equation 4, its diluted feature vector fails to align cleanly with any single class embedding during text-prompt classification.

We therefore distinguish the solver approximation error from the unavoidable semantic error of the least-squares target. By the orthogonality of least-squares residuals ($A^\top(A\widehat{X} - B) = 0$), the additional reconstruction error incurred by the one-shot estimator X' is defined by the *fitting gap*:

$$\mathcal{L}(X') - \mathcal{L}(\widehat{X}) = \frac{1}{2}\|A(X' - \widehat{X})\|^2 \leq \frac{1}{2}\sum_{i=1}^{R}\sum_{j,k} A_{ij}A_{ik}\|\widehat{x}_j - \widehat{x}_k\|_2^2. \tag{9}$$

The inequality additionally assumes complete absorption, $s_i = 1$, the same row-stochastic condition assumed by SFS's own approximation bound (Xiong et al., 2026). Here, $A_{ij}A_{ik}$ weights the co-contribution of primitives j and k along ray i, while $\|\widehat{x}_j - \widehat{x}_k\|_2^2$ captures their pairwise feature disagreement. This inequality shows that the fitting gap is bounded by the shared-ray feature disagreement between co-contributing primitives. Ray dispersion β (Eq. 7) provides a complementary ray-level measure of this contributor disagreement. Concentrating ray weights onto fewer primitives reduces these cross-contributor terms, hence reducing the upper bound on the fitting gap $\mathcal{L}(X') - \mathcal{L}(\widehat{X})$ and in extension allowing for better feature assignments.

However, because a small fitting gap cannot fix semantic ambiguity in the least-squares optimum $\widehat{X}$, we evaluate primitive purity P_j to measure whether the evidence aggregated at primitive j is dominated by a single class. Together, β, the fitting gap, and purity allow us to trace how the 3D reconstruction governs feature lifting: from how closely X' approximates $\widehat{X}$, to the class consistency of the evidence being aggregated, and ultimately to the accuracy of the recovered 3D labels. Figure 3 illustrates how spatial ownership, observation weights, and shared rays govern this interplay.

To disentangle these effects, we separate reconstruction-dependent errors from errors in the semantic observations serving as input to the lifting. We first isolate the geometric quality of the reconstruction by lifting ground-truth class masks encoded as text embeddings. Because both reconstructions receive perfect upstream labels, any difference in the resulting 3D labels proceeds from reconstruction and lifting process, unaffected by CLIP noise. We then evaluate open-vocabulary segmentation using real SAM/CLIP image features, testing whether the reconstruction-dependent effects persist under practical, noisy observation conditions.

4 Experiments

We evaluate the effect of the reconstruction on feature lifting at three levels. First, we compare the dispersion β and the fitting gap $\mathcal{L}(X') - \mathcal{L}(\widehat{X})$ to determine whether the observed differences in β correspond to differences in the quality of the one-shot lifting solution. We then examine purity and primitive-level prediction accuracy to determine whether the evidence accumulated by each reconstruction is more class-consistent.

To isolate the effect of the reconstruction from errors in the input image features, we next lift ground-truth image labels encoded as class text embeddings. This controlled experiment gives both reconstructions the same correct semantic observations and therefore tests whether differences in the recovered 3D labels arise from the reconstruction and lifting process itself. Finally, we evaluate CLIP image features, measuring whether the same differences persist when the lifted observations also contain errors.

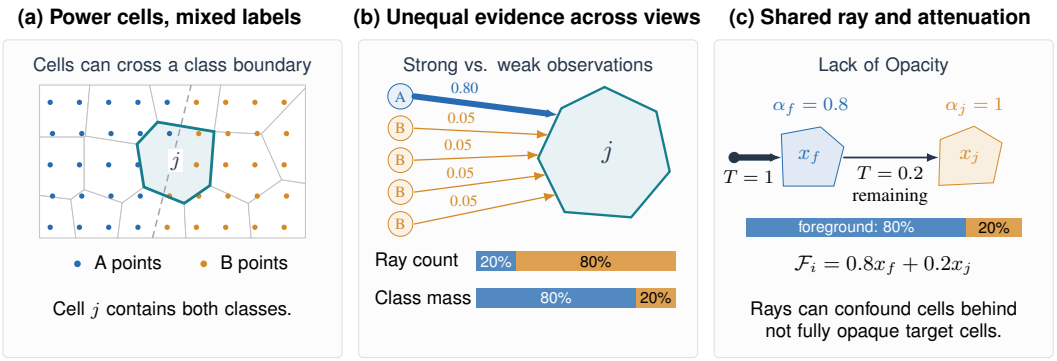


Figure 3: **How semantic evidence reaches a primitive.** (a) A cell can contain multiple classes. (b) Observation weights can reverse the majority suggested by ray counts. (c) Partial transparency lets foreground observations influence cells behind the surface.

4.1 Dispersion and Class Purity

We compare PowerFoam and 3DGS on ten ScanNet scenes (Dai et al., 2017), initializing both from the same GT vertices with one primitive per vertex, following the OpenGaussian (Wu et al., 2024) protocol. Centers and primitive counts remain fixed during reconstruction, and all reconstruction parameters remain fixed during lifting. We measure the number of nonzero contributors per ray, k_i, and dispersion β_i, using a least-squares reference computed on twelve matched views per scene. PSNR uses all views at native resolution.

Table 1: **Dispersion and ray contribution.** Ten-scene averages. Dispersion is the mean of scene medians. Per-scene results in Table 5.

Representation	PSNR ↑	k_i ↓	β_i ↓
PowerFoam	**27.01**	**2.32**	**0.0080**
3DGS	25.22	33.63	1.5554

Table 1 shows that PowerFoam uses fewer contributors per ray k_i and exhibits lower dispersion β_i while retaining higher average RGB reconstruction quality. To establish whether its one-shot estimate lies closer to least squares, we measure the fitting gap using GT image labels encoded as unit class text embeddings on twelve matched views. We compare raw, unnormalized feature fields against numerically converged, unregularized least-squares references. Averaging the per-ray gap $[\mathcal{L}(X') - \mathcal{L}(\widehat{X})]/R$ equally across scenes gives 4.663×10^{-3} for 3DGS and 8.352×10^{-4} for PowerFoam, an 82.1% reduction. PowerFoam has the smaller gap and lower one-shot loss in all ten scenes, although 3DGS reaches the lower least-squares reference loss. Thus, PowerFoam supports a more accurate one-shot approximation on these reconstructions, rather than a lower attainable least-squares loss.

Closer agreement with least squares does not establish whether a primitive receives consistent class evidence. We therefore measure purity from the evidence accumulated by the renderer and compare it with prediction accuracy when lifting CLIP features. Each observation contributes according to its rendering weight A_{ij}, and purity measures the fraction of accumulated class evidence belonging to the dominant class.

Figure 4 shows that accuracy increases with purity across all five bands for both representations. PowerFoam places more evaluated points in the highest-purity band, where predictions are most accurate. Its advantage therefore includes a more favorable distribution of incoming evidence, rather than consistently higher accuracy within every band. Purity tracks prediction accuracy even with real image features, but does not guarantee correct labels.

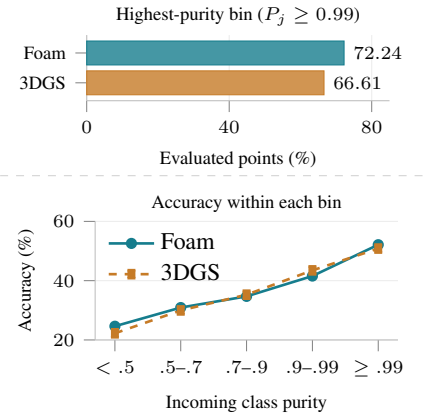


Figure 4: **CLIP feature purity.** Foam has more high-purity points. Higher purity tracks higher accuracy.

Even points assigned to the highest-purity primitives remain frequently misclassified with real CLIP features. Their contributing rays mostly observe the same GT class, but the image features need not

identify that class correctly. We therefore replace the image features with GT class embeddings to measure the segmentation errors that remain when the upstream labels are correct.

4.2 Lifting ground-truth semantics

To separate errors in the image features from errors introduced by reconstruction and lifting, we replace each labeled pixel's feature with the unit CLIP text embedding of its GT class, obtained from ScanNet's label masks. Unlabeled pixels contribute no evidence. With each reconstruction fixed, we apply Equation 5 over all available lifting views and classify the resulting primitive features by cosine similarity to the class text embeddings.

Table 2 shows that PowerFoam recovers more accurate point labels from the same correct image observations, with higher mIoU in all ten scenes. This advantage is not explained by a higher geometric upper bound. The geometric upper bound pinpoints the limit introduced by mixed labels, as described in Figure 3. Therefore, we conclude that PowerFoam recovers more of the label accuracy available because of improved assignment from the image-based lifting.

Table 2: **Lifting ground-truth image semantics.** The geometric upper bound (Geom UB) assigns each primitive the majority GT class of its evaluated points.

	GT feature lifting		Geom UB
Representation	Point accuracy	mIoU	mIoU
PowerFoam	**96.37**	**90.59**	98.13
3DGS	94.25	85.40	**98.92**

These results complement the fitting-gap measurement. PowerFoam's one-shot estimate is closer to its least-squares reference, and, when supplied with correct image labels, it also produces more accurate 3D segmentation. The remaining question is whether this advantage persists with imperfect image features and different aggregation methods. We evaluate that setting next by replacing the Gaussian representation in existing open-vocabulary segmentation methods with PowerFoam.

4.3 Open-vocabulary segmentation

We adapt existing Gaussian-based methods' feature lifting approaches to PowerFoam. Each method uses the same SAM/CLIP image features for both representations, allowing us to compare their segmentation results while holding the upstream observations fixed.

For ScanNet, we follow OpenGaussian's ten-scene point-level evaluation setup (Wu et al., 2024), with the changes to point assignment and opacity filtering described below. We assign each evaluated point its primitive's predicted class and report mIoU (mean intersection over union) and mAcc (mean class accuracy). Predictions use three different class sets. For PowerFoam, we assign points using the reconstructed surface inside each cell, represented by its dipole, rather than the cell center alone. This surface can shift during reconstruction even when the center remains fixed, so the center need not lie in the primitive's occupied region (Appendix B.1).

OpenGaussian's protocol excludes ground-truth points associated with Gaussians whose opacity is below 0.1. However, low-opacity primitives can still contribute to rendering and receive features during lifting. Excluding their associated points removes part of the reconstruction from assessment and allows the evaluated point set to depend on the representation. It also means that a higher score on retained points need not imply better performance on the full scene. We therefore retain all valid labeled ground-truth points for both representations, so weakly represented regions remain part of the segmentation task. Under this common point set, PowerFoam improves both metrics for every tested method across all three class sets (Table 3). For the opacity-filtered evaluation, see Table 6.

We also evaluate rendered text-query segmentation on Figurines, Ramen, Teatime, and Waldo kitchen from LERF (Kerr et al., 2023), following the SFS evaluation protocol (Xiong et al., 2026). For each query, we render the lifted features from the evaluation camera and compare them with the query's text embedding to obtain a segmentation mask. Both representations use the same query evaluator, reference masks, and threshold rules. Table 4 reports mean IoU and mean accuracy across the four scenes.

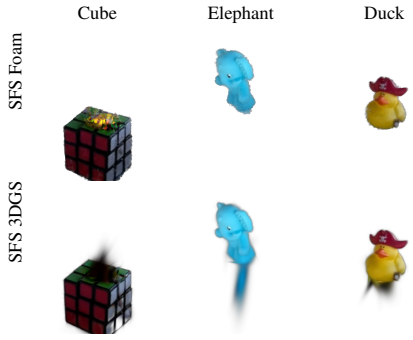


Table 3: **Open-vocabulary 3D segmentation on ScanNet.** We report mIoU and mAcc (%) for three class sets. Bold marks the higher score within each pair. Green parentheses show OV-Foam's improvement over 3DGS in percentage points. Appendix B details the protocol.

		19 classes		15 classes		10 classes	
Method	Representation	mIoU	mAcc	mIoU	mAcc	mIoU	mAcc
VALA (Wang et al., 2026)	3DGS	30.08	47.57	32.70	52.00	39.56	60.35
	OV-Foam	**33.68**	**51.18**	**37.15**	**56.08**	**45.68**	**65.39**
	Δ	+3.60	+3.61	+4.45	+4.08	+6.12	+5.04
Occam's LGS (Cheng et al., 2025)	3DGS	32.76	50.40	35.62	55.24	43.94	64.66
	OV-Foam	**34.19**	**52.34**	**37.58**	**57.46**	**46.46**	**66.64**
	Δ	+1.43	+1.94	+1.96	+2.22	+2.52	+1.98
SFS (Xiong et al., 2026)	3DGS	32.03	49.29	35.41	54.78	44.34	64.24
	OV-Foam	**34.00**	**51.75**	**37.59**	**57.43**	**46.91**	**66.64**
	Δ	+1.97	+2.46	+2.18	+2.65	+2.57	+2.40
LUDVIG (Marrie et al., 2025)	3DGS	32.94	50.36	36.58	56.22	45.64	65.43
	OV-Foam	**34.58**	**52.14**	**38.11**	**57.84**	**47.59**	**67.04**
	Δ	+1.64	+1.78	+1.53	+1.62	+1.95	+1.61

Table 4: **LERF text-query evaluation.** Mean IoU and mean accuracy across four scenes under a query and mask protocol. Bold marks the better representation. Δ denotes OV-Foam minus 3DGS.

	mIoU			mAcc		
Method	3DGS	OV-Foam	Δ	3DGS	OV-Foam	Δ
VALA	0.555	**0.570**	+0.015	0.707	**0.730**	+0.023
Occam's LGS (Cheng et al., 2025)	0.522	**0.529**	+0.007	0.699	**0.749**	+0.050
SFS (Xiong et al., 2026)	0.502	**0.570**	+0.068	0.641	**0.659**	+0.018
LUDVIG (Marrie et al., 2025)	0.438	**0.543**	+0.105	0.735	**0.746**	+0.011

OV-Foam improves both metrics for all four methods, extending the advantage from 3D point labels to rendered segmentation masks. Together, these results show that PowerFoam's advantage extends beyond correct upstream labels and a single lifting method. The earlier measurements identify differences in how the representations distribute and combine image evidence. The CLIP feature benchmarks establish that replacing Gaussians with PowerFoam can turn those representation differences into better open-vocabulary segmentation, even when the input features remain imperfect.

5 Limitations and Conclusion

Our ScanNet experiments cover ten scenes and initialize both representations from the same point cloud through the SfM initialization pipeline. For this benchmark, the supplied point cloud is the ground-truth geometry provided by ScanNet rather than an image-derived SfM reconstruction. The ground-truth lifting experiment isolates errors that remain with correct image labels, but does not establish an upper bound over all lifting methods. Our fitting-gap analysis concerns feature reconstruction loss rather than segmentation accuracy, and high purity can still reflect consistently incorrect observations. The observed advantages therefore depend on the reconstruction, input features, and evaluation setting.

We investigated how the choice of reconstruction primitives affects feature lifting. Under matched reconstruction budgets, PowerFoam has fewer contributors per ray and lower measured dispersion than 3DGS, while placing more evaluated points on primitives with high-purity evidence. Its one-shot estimate is closer to its least-squares reference, and lifting correct image labels produces more accurate 3D segmentation. These advantages extend to real image features, adapting existing lifting methods to PowerFoam improves the tested ScanNet and LERF segmentation results. Together, these findings establish representation choice as a practical way to improve semantic lifting without requiring more accurate upstream image features.

AI USE STATEMENT

In this work, we used generative AI tools for drafting and editing sections of the paper. We also used AI tools to find related works beyond what our manual research found. We used AI models to review and critique our paper and used its feedback to improve our work. AI coding agents were used for parts of the implementation and debugging. All generated text, code and suggested papers were reviewed thoroughly by the authors. In all cases we used AI tools as *tools* to assist our work, never to replace or automate full aspects of the research process.

REPRODUCIBILITY STATEMENT

With our paper, we release the full source code required to reproduce our results. We will upload the reconstructed 3DGS and Powerfoam scenes and the lifted feature sets used as input for our segmentation. We will supply a simple Python script that reproduces the segmentations that lead to the results.

REFERENCES

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 *Proceedings of the 42nd International Conference on Machine Learning*, volume 267 of *Proceedings of Machine Learning Research*, pp. 7013–7034. PMLR, 2025. URL https://proceedings.mlr.press/v267/cen25a.html.

Jiahuan Cheng, Jan-Nico Zaech, Luc Van Gool, and Danda Pani Paudel. Occam's LGS: An efficient approach for language gaussian splatting. In *Proceedings of the 36th British Machine Vision Conference*. BMVA, 2025. URL https://bmvc2025.bmva.org/proceedings/694/.

Angela Dai, Angel X. Chang, Manolis Savva, Maciej Halber, Thomas Funkhouser, and Matthias Nießner. ScanNet: Richly-annotated 3D reconstructions of indoor scenes. In *IEEE Conference on Computer Vision and Pattern Recognition*, pp. 5828–5839, 2017. URL https://openaccess.thecvf.com/content_cvpr_2017/html/Dai_ScanNet_Richly-Annotated_3D_CVPR_2017_paper.html.

Shaohui Dai, Yansong Qu, Zheyan Li, Xinyang Li, Shengchuan Zhang, and Liujuan Cao. Training-free hierarchical scene understanding for gaussian splatting with superpoint graphs. In *Proceedings of the 33rd ACM International Conference on Multimedia*, pp. 3673–3682. Association for Computing Machinery, 2025. doi: 10.1145/3746027.3755146. URL https://doi.org/10.1145/3746027.3755146.

Bin Dou, Tianyu Zhang, Zhaohui Wang, Yongjia Ma, Zejian Yuan, and Nanning Zheng. Learning segmented 3D gaussians via efficient feature unprojection for zero-shot neural scene segmentation. In *Neural Information Processing – ICONIP 2024*, volume 15293 of *Lecture Notes in Computer Science*, pp. 398–412. Springer, 2025. doi: 10.1007/978-981-96-6596-9_28. URL https://link.springer.com/chapter/10.1007/978-981-96-6596-9_28.

Shrisudhan Govindarajan, Daniel Rebain, Kwang Moo Yi, and Andrea Tagliasacchi. Radiant foam: Real-time differentiable ray tracing. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 4135–4145, 2025. URL https://openaccess.thecvf.com/content/ICCV2025/html/Govindarajan_Radiant_Foam_Real-Time_Differentiable_Ray_Tracing_ICCV_2025_paper.html.

Shrisudhan Govindarajan, Daniel Rebain, Dor Verbin, Kwang Moo Yi, Anish Prabhu, and Andrea Tagliasacchi. Power foam: Unifying real-time differentiable ray tracing and rasterization. *arXiv preprint arXiv:2604.24994*, 2026. URL https://arxiv.org/abs/2604.24994.

Qiao Gu, Ali Kuwajerwala, Sacha Morin, Krishna Murthy Jatavallabhula, Bipasha Sen, Aditya Agarwal, Corban Rivera, William Paul, Kirsty Ellis, Rama Chellappa, Chuang Gan, Celso Miguel de Melo, Joshua B. Tenenbaum, Antonio Torralba, Florian Shkurti, and Liam Paull. ConceptGraphs: Open-vocabulary 3D scene graphs for perception and planning. In *2024 IEEE International Conference on Robotics and Automation (ICRA)*, pp. 5021–5028. IEEE, 2024. doi:

10.1109/ICRA57147.2024.10610243. URL https://doi.org/10.1109/ICRA57147.2024.10610243.

Kim Jun-Seong, GeonU Kim, Kim Yu-Ji, Yu-Chiang Frank Wang, Jaesung Choe, and Tae-Hyun Oh. Dr. Splat: Directly referring 3D gaussian splatting via direct language embedding registration. In *2025 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 14137–14146. IEEE, 2025. doi: 10.1109/CVPR52734.2025.01319. URL https://doi.org/10.1109/CVPR52734.2025.01319.

Bernhard Kerbl, Georgios Kopanas, Thomas Leimkühler, and George Drettakis. 3D gaussian splatting for real-time radiance field rendering. *ACM Transactions on Graphics*, 42(4):1–14, 2023. doi: 10.1145/3592433. URL https://repo-sam.inria.fr/fungraph/3d-gaussian-splatting/.

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*, pp. 19729–19739, 2023. URL https://openaccess.thecvf.com/content/ICCV2023/html/Kerr_LERF_Language_Embedded_Radiance_Fields_ICCV_2023_paper.html.

Alexander Kirillov, Eric Mintun, Nikhila Ravi, Hanzi Mao, Chloe Rolland, Laura Gustafson, Tete Xiao, Spencer Whitehead, Alexander C. Berg, Wan-Yen Lo, Piotr Dollár, and Ross Girshick. Segment anything. In *IEEE/CVF International Conference on Computer Vision*, pp. 4015–4026, 2023. URL https://openaccess.thecvf.com/content/ICCV2023/html/Kirillov_Segment_Anything_ICCV_2023_paper.html.

Juliette Marrie, Romain Menegaux, Michael Arbel, Diane Larlus, and Julien Mairal. LUDVIG: Learning-free uplifting of 2D visual features to gaussian splatting scenes. In *IEEE/CVF International Conference on Computer Vision*, pp. 7440–7450, 2025. URL https://openaccess.thecvf.com/content/ICCV2025/html/Marrie_LUDVIG_Learning-Free_Uplifting_of_2D_Visual_Features_to_Gaussian_Splatting_ICCV_2025_paper.html.

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*, pp. 20051–20060, 2024. URL https://openaccess.thecvf.com/content/CVPR2024/html/Qin_LangSplat_3D_Language_Gaussian_Splatting_CVPR_2024_paper.html.

Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, Gretchen Krueger, and Ilya Sutskever. Learning transferable visual models from natural language supervision. In *Proceedings of the 38th International Conference on Machine Learning*, volume 139 of *Proceedings of Machine Learning Research*, pp. 8748–8763. PMLR, 2021a. URL https://proceedings.mlr.press/v139/radford21a.html.

Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, Gretchen Krueger, and Ilya Sutskever. Learning transferable visual models from natural language supervision. In *Proceedings of the 38th International Conference on Machine Learning*, volume 139 of *Proceedings of Machine Learning Research*, pp. 8748–8763. PMLR, 2021b. URL https://proceedings.mlr.press/v139/radford21a.html.

Amr Sharafeldin, Aryan Mikaeili, Thomas Walker, Shrisudhan Govindarajan, Daniel Rebain, Kwang Moo Yi, and Andrea Tagliasacchi. Semantic foam: Unifying spatial and semantic scene decomposition. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 29814–29823, 2026. URL https://openaccess.thecvf.com/content/CVPR2026/html/Sharafeldin_Semantic_Foam_Unifying_Spatial_and_Semantic_Scene_Decomposition_CVPR_2026_paper.html.

Haoyu Wang, Sizheng Guo, Jin Ye, Zhongying Deng, Junlong Cheng, Tianbin Li, Jianpin Chen, Yanzhou Su, Ziyan Huang, Yiqing Shen, Bin Fu, Shaoting Zhang, and Junjun He. SAM-Med3D: A vision foundation model for general-purpose segmentation on volumetric medical images. *IEEE Transactions on Neural Networks and Learning Systems*, 36(10):17599–17612, 2025. doi: 10.1109/TNNLS.2025.3586694. URL https://doi.org/10.1109/TNNLS.2025.3586694.

Sen Wang, Kunyi Li, Siyun Liang, Elena Alegret, Jing Ma, Nassir Navab, and Stefano Gasperini. Visibility-aware language aggregation for open-vocabulary segmentation in 3D gaussian splatting. In *2026 International Conference on 3D Vision (3DV)*, pp. 1791–1800, 2026. doi: 10.1109/3DV69130.2026.00170. URL https://doi.org/10.1109/3DV69130.2026.00170.

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*, volume 37, 2024. doi: 10.52202/079017-0604. URL https://proceedings.neurips.cc/paper_files/paper/2024/hash/21f7b745f73ce0d1f9bcea7f40b1388e-Abstract-Conference.html.

Butian Xiong, Rong Liu, Kenneth Xu, Meida Chen, and Andrew Feng. Splat feature solver. In *International Conference on Learning Representations*, 2026. URL https://arxiv.org/abs/2508.12216.

A Dispersion and Contribution count

Table 5: **Per-scene matched-budget results on ScanNet.** Both representations use the same primitive count P within each scene. PSNR uses all views. Contributor count k_i and dispersion β_i use twelve views per scene. The final row averages the scene summaries, including the scene medians of β_i. Bold indicates higher PSNR or lower contributor count and dispersion.

		PSNR (dB)		Mean k_i		Median β_i	
Scene	P	Foam	3DGS	Foam	3DGS	Foam	3DGS
scene0000_00	81,369	**21.84**	19.74	**2.32**	31.52	**0.0053**	1.8443
scene0062_00	51,610	**27.31**	23.16	**2.02**	33.90	**0.0000**	1.0508
scene0070_00	109,380	28.13	**29.46**	**2.69**	34.11	**0.0318**	2.1095
scene0097_00	72,007	31.94	**34.44**	**1.91**	33.26	**0.0000**	1.1392
scene0140_00	372,941	**27.25**	26.62	**2.76**	34.24	**0.0092**	1.6057
scene0200_00	83,291	**26.06**	21.09	**2.00**	34.82	**0.0000**	1.1613
scene0347_00	67,984	**25.24**	20.56	**2.13**	31.35	**0.0001**	1.2667
scene0400_00	155,959	**26.60**	21.48	**1.96**	36.26	**0.0000**	1.5230
scene0590_00	222,957	31.67	**32.50**	**2.73**	33.52	**0.0035**	1.7131
scene0645_00	352,477	**24.04**	23.15	**2.66**	33.36	**0.0302**	2.1408
Mean	156,998	**27.01**	25.22	**2.32**	33.63	**0.0080**	1.5554

B ScanNet evaluation protocol

We follow OpenGaussian's ten-scene point-level evaluation setup (Wu et al., 2024). For our matched reconstructions, both representations start from the same ground-truth vertices, with one primitive per vertex. Primitive centers and counts remain fixed during reconstruction, while the remaining reconstruction parameters are optimized. We then hold the reconstruction fixed during feature lifting.

We assign semantic labels by maximum cosine similarity to the class text embeddings and report mIoU and mAcc for the 19-, 15-, and 10-class evaluations. Within each scene, these metrics average class IoU and class recall over the valid ground-truth classes present. Final scores average the scene results.

B.1 Dipole-aware point assignment

For OV-Foam, point ownership uses the reconstructed dipole geometry. A fixed initialization center need not lie on the occupied side of its primitive's displaced surface. For each evaluation point q, we take the 64 Euclidean-nearest supported primitive centers and rank them by power distance $\|q - p_j\|^2 - a_j^2$. We select the first candidate satisfying $(q - p_j) \cdot n_j \leq h_j(q)$, where $h_j(q)$ is the interpolated dipole height at the point's tangent-plane coordinates. This test checks the occupied side of the dipole surface.

If no candidate passes, we use the minimum power distance in the same pool. The fallback preserves assignment coverage, so the dipole test does not discard points. Power distances are evaluated in double precision. The finite candidate search approximates global power-distance ranking.

B.2 Evaluation without opacity culling

OpenGaussian excludes GT points associated with Gaussians whose opacity is below 0.1. The Del.% column in Table 6 reports the fraction of labeled points excluded before scoring. Rates marked ‡ come from our matched reproductions. This exclusion changes which points contribute to the metrics. To compare both representations on the same evaluation support, Table 3 retains all valid labeled ground-truth points, keeping the reconstructions and lifted features fixed. Removing opacity culling widens OV-Foam's advantage over the matched Gaussian baselines. OV-Foam also achieves higher mIoU and mAcc for every method across the class set evaluations. These results show that its advantage persists and increases when low-opacity points are included in scoring.

Table 6: **ScanNet results under opacity-filtered evaluation.** Published Gaussian scores are paired with our OV-Foam adaptations. Bold marks the better score within each pair. Green parentheses show OV-Foam minus 3DGS in percentage points. Del.% denotes excluded labeled points. ‡Rates are measured on our matched reproductions, not reported by the cited papers. Table 3 reports results without filtering.

		19 classes		15 classes		10 classes		
Method	Representation	mIoU	mAcc	mIoU	mAcc	mIoU	mAcc	Del.%
VALA (Wang et al., 2026)	3DGS	32.11	50.05	35.10	54.77	**46.21**	**65.61**	14.16‡
	OV-Foam	(+1.57) **33.68**	(+1.13) **51.18**	(+2.05) **37.15**	(+1.31) **56.08**	(−0.53) 45.68	(−0.22) 65.39	**0.00**
Occam's LGS (Cheng et al., 2025)	3DGS	31.93	48.93	34.25	53.71	45.16	64.39	19.28‡
	OV-Foam	(+2.26) **34.19**	(+3.42) **52.35**	(+3.34) **37.59**	(+3.76) **57.47**	(+1.31) **46.47**	(+2.26) **66.65**	**0.10**
SFS (Xiong et al., 2026)	3DGS	33.33	51.35	36.43	55.38	44.74	63.53	12.66‡
	OV-Foam	(+0.68) **34.01**	(+0.41) **51.76**	(+1.18) **37.61**	(+2.07) **57.45**	(+2.18) **46.92**	(+3.13) **66.66**	**0.10**
LUDVIG (Marrie et al., 2025)	3DGS	33.90	51.40	37.40	57.20	46.40	66.20	15.92‡
	OV-Foam	(+0.69) **34.59**	(+0.75) **52.15**	(+0.72) **38.12**	(+0.66) **57.86**	(+1.20) **47.60**	(+0.85) **67.05**	**0.11**

C Full Oracle table

Table 7: **Per-scene lifting of ground-truth image semantics on ScanNet.** Point accuracy and mIoU (%) when each valid pixel carries its ground-truth class text embedding, using ScanNet's released 2D label masks. Geom UB is the geometric upper bound: each primitive is assigned the majority ground-truth class of its evaluated points. Bold marks the higher score within each pair. C is the number of ground-truth classes present, and N is the number of scored points. This shows 3DGS doesn't lose due to lack of capacity but due to the solve errors. The Mean row is the equal-weight average over scenes and matches Table 2.

			Point accuracy			mIoU			Geom UB (mIoU)		
Scene	C	N	3DGS	OV-Foam	Δ	3DGS	OV-Foam	Δ	3DGS	OV-Foam	Δ
scene0000_00	14	54,480	92.98	**96.10**	(+3.12)	83.61	**91.27**	(+7.66)	97.00	**97.81**	(+0.81)
scene0062_00	6	41,873	95.85	**96.50**	(+0.65)	86.53	**87.82**	(+1.29)	**98.94**	97.50	(-1.44)
scene0070_00	9	69,922	94.10	**96.77**	(+2.67)	83.30	**89.89**	(+6.59)	**98.69**	97.92	(-0.77)
scene0097_00	7	52,656	95.41	**97.34**	(+1.93)	88.37	**93.06**	(+4.69)	**99.33**	97.81	(-1.52)
scene0140_00	8	279,359	91.82	**94.11**	(+2.29)	86.03	**90.80**	(+4.77)	**99.19**	98.95	(-0.24)
scene0200_00	8	65,082	93.65	**96.07**	(+2.42)	83.11	**89.59**	(+6.48)	**99.20**	97.69	(-1.51)
scene0347_00	7	48,063	95.06	**97.07**	(+2.01)	88.44	**93.10**	(+4.66)	**99.34**	99.05	(-0.29)
scene0400_00	7	93,930	94.94	**97.26**	(+2.32)	85.36	**91.23**	(+5.87)	**99.57**	98.87	(-0.70)
scene0590_00	14	153,151	93.91	**95.78**	(+1.87)	84.74	**88.85**	(+4.11)	**98.99**	97.28	(-1.71)
scene0645_00	16	253,034	94.76	**96.71**	(+1.95)	84.55	**90.24**	(+5.69)	**98.96**	98.46	(-0.50)
Mean	–	–	94.25	**96.37**	(+2.12)	85.40	**90.59**	(+5.19)	**98.92**	98.13	(-0.79)

D Proof of the fitting-gap identity

Let A contain the rendering weights, B the observed image features, and X the primitive features. The rendering loss is

$$\mathcal{L}(X) = \frac{1}{2}\|AX - B\|_F^2 = \frac{1}{2}\sum_{i=1}^{R}\sum_{f=1}^{d}\big((AX)_{if} - B_{if}\big)^2, \tag{10}$$

where R is the number of rays and d is the feature dimension. The subscript F denotes the Frobenius norm, which sums squared entries of a matrix as shown above.

Let $\widehat{X}$ be an exact, unconstrained least-squares minimizer and X' the one-shot estimate, both using the same A and B. Then

$$\boxed{\mathcal{L}(X') - \mathcal{L}(\widehat{X}) = \frac{1}{2}\|A(X' - \widehat{X})\|_F^2.} \tag{11}$$

To prove this, write the one-shot rendering residual as

$$AX' - B = (A\widehat{X} - B) + A(X' - \widehat{X}).$$

Expanding its squared norm gives

$$\mathcal{L}(X') - \mathcal{L}(\widehat{X}) = \frac{1}{2}\|A(X' - \widehat{X})\|_F^2 \tag{12}$$

$$+ \sum_{i=1}^{R}\sum_{f=1}^{d}(A\widehat{X} - B)_{if}\left[A(X' - \widehat{X})\right]_{if}. \tag{13}$$

The final term is zero because the least-squares normal equations give $A^\top(A\widehat{X} - B) = 0$. Thus the fitting gap equals half the squared difference between the image features rendered by the two estimates.

The identity allows incomplete absorption and does not require a unique least-squares solution. It measures agreement between rendered features, not correctness of the resulting semantic labels. For a numerically computed reference, the equality holds up to its remaining optimization error.

E QUALITATIVE FIGURES

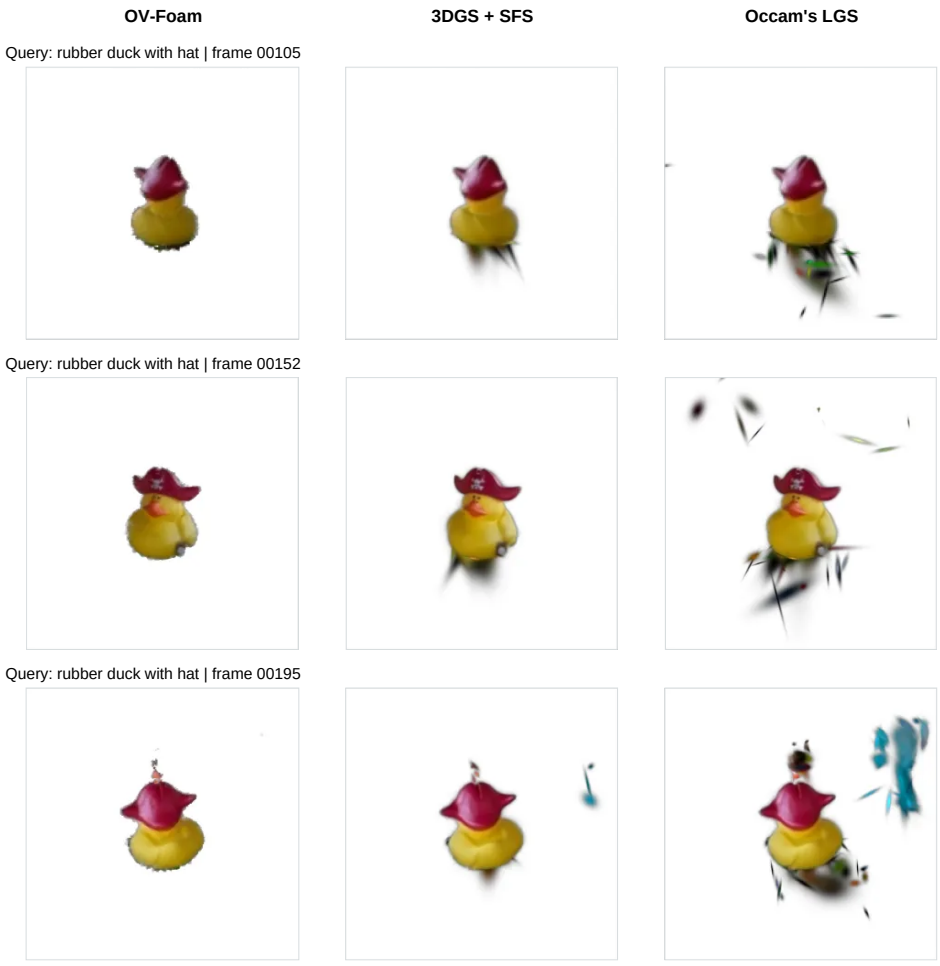


Figure 5: Object extraction for “rubber duck with hat, frames 00105, 00152 and 00195. OV-Foam uses SFS here



Figure 6: Object extraction for “rubics cube, frames 00105, 00152 and 00195. OV-Foam uses SFS here

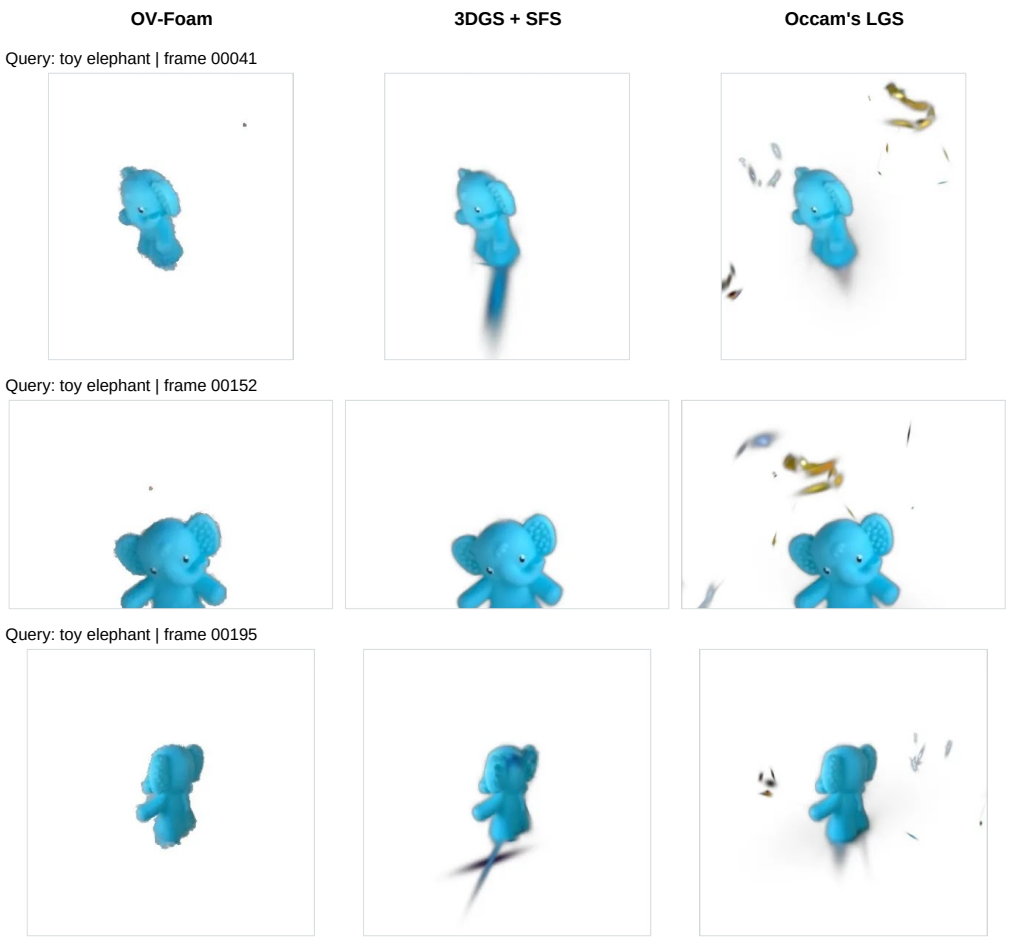


Figure 7: Object extraction for “toy elephant, frames 00041, 00152 and 00195. OV-Foam uses SFS here

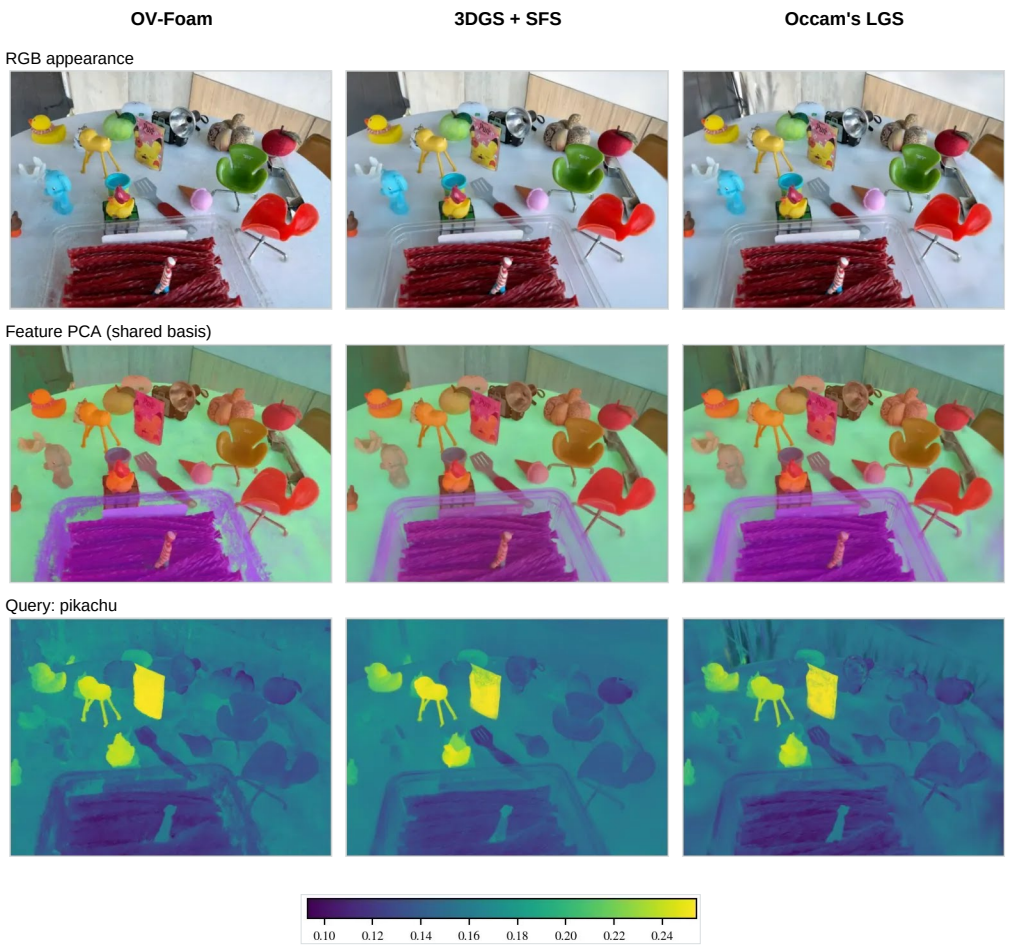


Figure 8: RGB appearance, shared-basis feature PCA and cosine similarity to “pikachu, frame 00041. OV-Foam uses SFS here

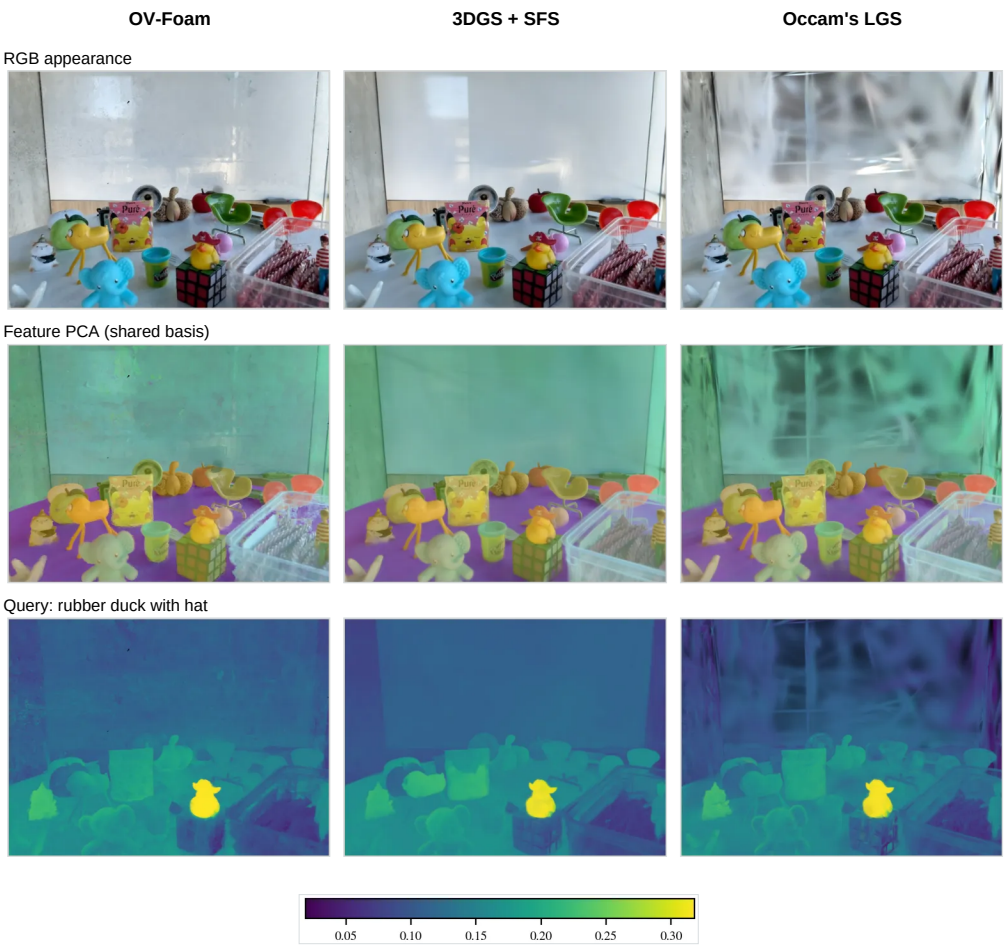


Figure 9: RGB appearance, shared-basis feature PCA and cosine similarity to “rubber duck with hat, frame 00152. The supplied overlay uses alpha 0.55. OV-Foam uses SFS here

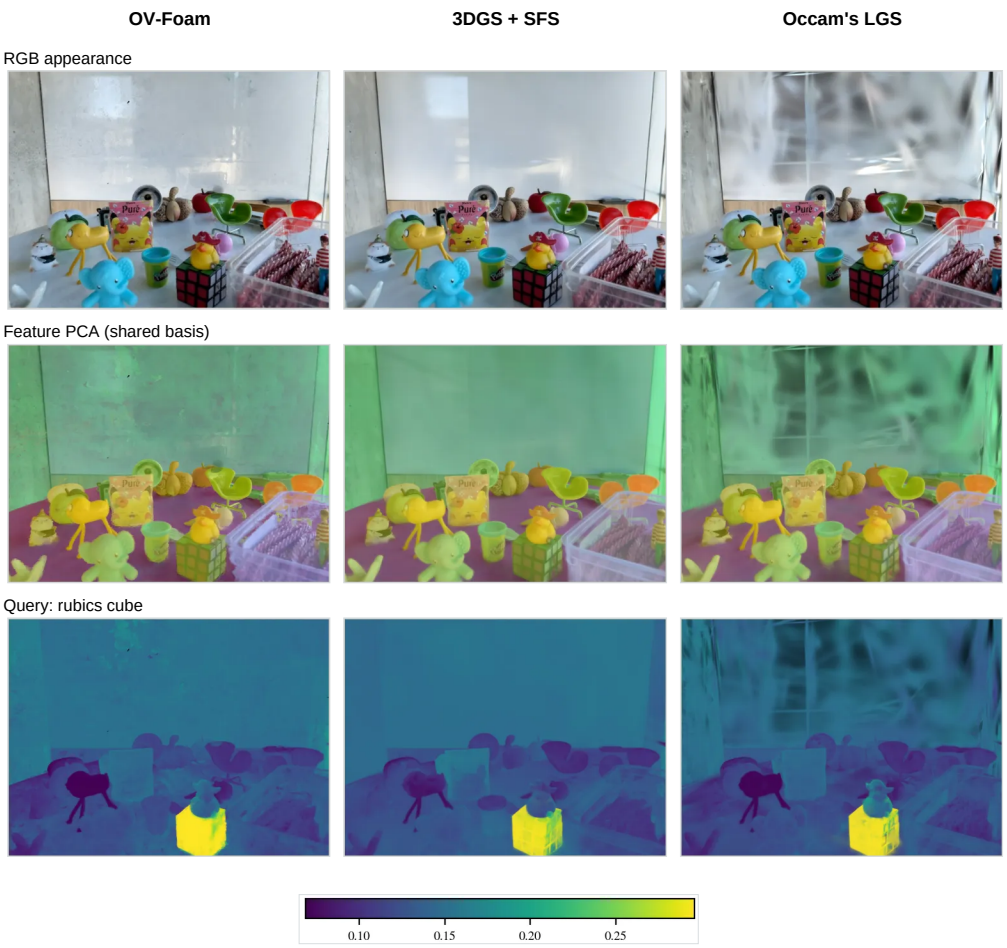


Figure 10: RGB appearance, shared-basis feature PCA and cosine similarity to “rubics cube, frame 00152. The supplied overlay uses alpha 0.55. OV-Foam uses SFS here