VERDICTGS: VIEWS AS NOISY ANNOTATORS FOR QUERY-TIME OPEN-VOCABULARY GAUSSIAN SPLATTING SEGMENTATION

Anonymous authors
Paper under double-blind review

ABSTRACT

Existing methods for open-vocabulary segmentation with 3D Gaussian Splatting integrate multi-view language features into 3D representations before any query. This adds aggregation, registration, and often grouping costs; moreover, query-independent aggregation mixes observations before their relevance to the query is known, predefined groups constrain the target's extent, and thresholds must be tuned before the query yet still ignore query-specific score distributions. We propose VerdictGS, a query-time inference framework that builds a 3D relevance field directly from individual observations' responses to a text query. Treating each view as a noisy annotator, VerdictGS aggregates per-view text–mask responses onto Gaussians and fits a per-query two-population binomial mixture to them with EM and without labels, yielding membership evidence whose decision boundary adapts to the query; this evidence is then fused across mask granularities. Because every decision is made at query time, VerdictGS builds no language representation before the query, lets each query weigh the original observations rather than their mixture, determines the target extent per Gaussian rather than within predefined groups, and needs no threshold tuning. Reducing this inference to sparse matrix operations and low-dimensional model fits, VerdictGS prepares scenes faster than training-free baselines and answers queries in comparable time, while achieving strong segmentation accuracy under each benchmark's evaluation.

1 INTRODUCTION

Open-vocabulary 3D segmentation enables users to segment objects, parts, and regions in reconstructed scenes through natural-language queries. The explicit scene representation and efficient rendering of 3D Gaussian Splatting (3DGS) provide a practical foundation for this task (Kerbl et al., 2023). Recent methods equip Gaussian scenes with language understanding by transferring language-aligned features from multi-view images into 3D. However, this transfer presents several challenges: high-dimensional features incur substantial computation and storage, occlusion can cause incorrect feature assignment, and variations in viewpoint and mask granularity introduce inconsistent semantic observations. Representing both entire objects and their constituent parts further requires accommodating different levels of spatial granularity.

One line of work equips individual Gaussians with language features through per-scene optimization (Qin et al., 2024) or training-free aggregation and registration of multi-view features (Cheng et al., 2025; Jun-Seong et al., 2025; Wang et al., 2026; Xiong et al., 2026). Another line organizes Gaussians into object groups or hierarchies and associates them with language features or semantic descriptions (Dai et al., 2025; Xie et al., 2026; Ding et al., 2026).

Despite different formulations, these approaches share a design we call pre-query aggregation: multi-view observations are integrated into a scene-associated language representation before a query is processed. Subsequent query evaluation operates on the resulting features and, where applicable, predefined object groups. These pipelines face three limitations.

First, constructing these representations adds a language-specific stage to every scene. Beyond extracting image masks and language features, methods aggregate and register high-dimensional features on the Gaussians, filter observations, group Gaussians, or build semantic indices, often on a

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.

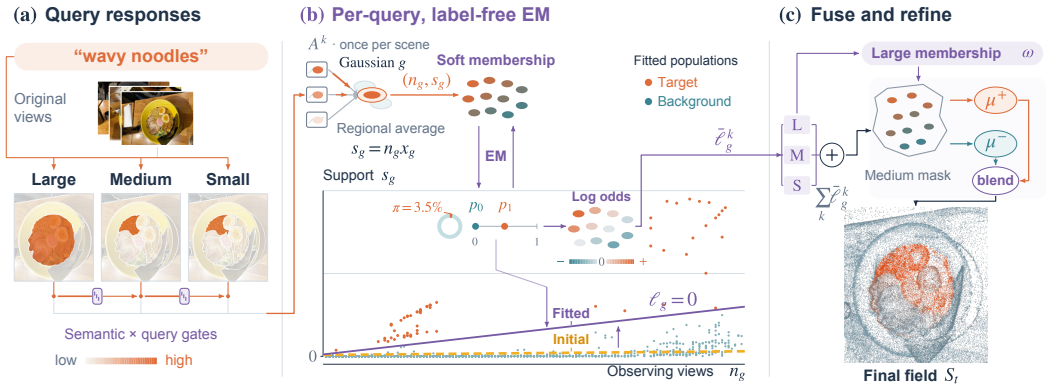

Figure 1: **Overview of VerdictGS.** (a) Masks compete for the query within each view, and the responses are gated across mask levels. (b) Fixed correspondences A^k and regional averaging give each Gaussian its effective support s_g over n_g observing views. Per-query, per-level EM estimates memberships, response rates (markers), and the target proportion (ring), yielding Gaussian log odds (color: sign and magnitude) and an adapted boundary (dashed: initial; solid: fitted). (c) These log odds are fused and refined by membership-conditioned averaging. Connection, gating, membership, score, and averaging sketches are schematic.

GPU, and some additionally invoke a multimodal large language model (MLLM) to describe object groups. These representations can be reused across queries, but every scene must pass through this stage before it can be queried.

Second, once observations are aggregated into features or assigned to groups before the query, this mixing cannot be undone at query time. Masks from different views may capture an entire object, a part, or surrounding context. Combining them into representative features mixes their semantic content before relevance to the current query can be assessed; matching only these features cannot reweight the original observations. Likewise, selection from predefined groups cannot recover a target whose extent cannot be expressed by their available combinations.

Third, relevance scores alone do not determine target extent. Converting scores into a region requires a threshold chosen before the query, often by tuning on benchmark annotations: a fixed value, a fraction of each query's score range, or a number of top-scoring groups. As score distributions vary across queries, the same value, fraction, or number can over-select some targets and under-select others.

We therefore ask: **can the target of a query be determined directly from multi-view observations once the query is given?** To address this agenda, we propose **VerdictGS**, which answers this question by treating each view as a noisy annotator of target membership. Instead of pre-query aggregation, VerdictGS evaluates each view at query time (Fig. 1(a)): within the view, masks compete for the query against a null reference, so the view's responses indicate which region matches the query, or that none does. These per-view responses, rather than features, are then accumulated on Gaussians through fixed rendering correspondences at every mask granularity, so the extent of the target, from a whole container to a small item inside it, is decided by the query rather than by predefined groups. Under identical inputs, accumulating responses instead of features improves segmentation on every benchmark.

The accumulated responses still need to be turned into a decision. Instead of applying a threshold chosen before the query, VerdictGS infers target membership for each query by fitting a two-population binomial mixture, a latent-class model as in label aggregation (Dawid & Skene, 1979), to the accumulated responses with the expectation–maximization (EM) algorithm (Dempster et al., 1977) and without labels. The sign of its log posterior odds gives the verdict on membership, so the decision boundary adapts to the query (Fig. 1(b)). Without labels or manual tuning, this boundary segments close to the best fixed threshold selected with ground truth, and the mixture also raises segmentation under the benchmark evaluations. Since responses are computed and combined at query time, VerdictGS produces a 3D relevance field over the Gaussians for each query without language-feature optimization, registration, or grouping beforehand, and its inference reduces to sparse matrix operations and low-dimensional model fits on cached observations.

Our contributions are summarized as follows:

- **Query-time inference without a pre-query language representation.** We propose VerdictGS, which infers a query-specific 3D relevance field from multi-view observations once the query is given, instead of building per-Gaussian language features or object groups for each scene beforehand.
- **Views as noisy annotators of Gaussian membership.** To avoid the irreversible mixing of observations in pre-query feature aggregation, we evaluate each view against the query before aggregation, treat its responses as noisy annotations of Gaussian membership, and aggregate them with a per-query latent-class binomial mixture fitted by EM without labels, which improves segmentation over feature aggregation under identical inputs.
- **Label-free decision boundary.** The per-query mixture provides a decision boundary that adapts to each query without fixed or benchmark-tuned thresholds, and we demonstrate that, without labels, it segments close to the best ground-truth-selected threshold.
- **Efficient query-time inference.** VerdictGS reduces per-query inference to sparse matrix operations and low-dimensional model fits, preparing scenes faster than methods that build language representations and answering queries in comparable time with strong accuracy under each benchmark's official evaluation.

2 Related Work

Learning language fields. LERF (Kerr et al., 2023) distills multi-scale CLIP features into a radiance field, and LangSplat (Qin et al., 2024) brings SAM-masked CLIP features to Gaussians through a scene-specific autoencoder; both optimize a semantic representation for each scene before any query.

Training-free feature lifting. Training-free methods lift multi-view features onto Gaussians directly: Occam's LGS (Cheng et al., 2025) derives a rendering-weighted aggregation rule, Dr. Splat (Jun-Seong et al., 2025) registers mask embeddings compressed with product quantization, VALA (Wang et al., 2026) gates assignments by visibility, and Splat Feature Solver (Xiong et al., 2026) solves lifting as a sparse linear inverse problem. They refine which observations contribute and how features are combined, but queries are still matched against the aggregated Gaussian features, whereas VerdictGS evaluates each observation against the query before accumulating them.

Grouping and hierarchical association. THGS (Dai et al., 2025) builds a superpoint hierarchy from geometric and mask cues before any query, ReLaGS (Xie et al., 2026) refines it with pruning and scene-graph relations, and ExtrinSplat (Ding et al., 2026) groups tracked masks and indexes them with MLLM-generated descriptions. Queries then select among these fixed regions, whereas VerdictGS scores each Gaussian, so the target extent is decided at query time.

From relevance scores to target regions. These methods select regions with a threshold chosen before the query: a value selected on the evaluation annotations, for the method or its baselines (Jun-Seong et al., 2025; Ding et al., 2026; Zang et al., 2026; Cha et al., 2026), a rule that changes across benchmarks (Cheng et al., 2025), a cut on scores normalized within each query (Wang et al., 2026), or a preset fraction of the maximum score or number of top-scoring groups (Peng et al., 2026; Dai et al., 2025; Lu et al., 2025). None of these rules reflects how each query's scores divide into target and background. Label aggregation jointly estimates hidden labels and annotator reliability from noisy annotations with EM (Dawid & Skene, 1979; Warfield et al., 2004); VerdictGS treats views as annotators of Gaussian membership, with response rates shared across views and refitted per query, so its decision boundary follows the query.

3 Method

VerdictGS infers a 3D relevance field by evaluating individual observations against a text query before aggregating their responses. We first describe the single-level formulation (Secs. 3.1–3.3), then extend it to multiple observation granularities (Sec. 3.4; Fig. 1).

3.1 Problem Formulation and Observation–Gaussian Association

We retain observations for query-time evaluation by storing their geometric correspondences separately from their language features. Our experiments use reconstructed 2D Gaussian Splatting (2DGS) scenes (Huang et al., 2024). Let $\mathcal{G} = \{G_g\}_{g=1}^{N}$ denote their Gaussians and $\mathcal{V}$ the calibrated training views. Each view provides cached SAM masks (Kirillov et al., 2023) and normalized CLIP features (Radford et al., 2021) in the four-level LangSplat preprocessing format (Qin et al., 2024): $\mathcal{K} = \{\text{default}, \text{small}, \text{medium}, \text{large}\}$, where default mixes the granularities of SAM's three mask outputs (Appendix B.1). Given a normalized text embedding t, we infer a scalar field S_t over the Gaussians.

For each level k, a sparse matrix A^k associates mask m with Gaussian g through its rendering contribution (Jun-Seong et al., 2025):

$$\begin{aligned} A^k_{gm} &= \sum_{\mathbf{u}\in m} \alpha_g^{v(m)}(\mathbf{u}) T_g^{v(m)}(\mathbf{u}), \\ \mathcal{L}_k(y) &= (D^k_G)^{-1} A^k y, \qquad \mathcal{R}_k(x) = (D^k_M)^{-1} (A^k)^\top x. \end{aligned} \tag{1}$$

Here $v(m)$ is the source view, $\alpha_g^v(\mathbf{u})$ is the effective opacity evaluated at the ray–splat intersection, and $T_g^v(\mathbf{u})$ is the preceding transmittance. The diagonal normalizers contain the row and column sums of A^k. Thus, $\mathcal{L}_k$ lifts mask values to Gaussians and $\mathcal{R}_k$ averages Gaussian values over mask support. These correspondences are prepared once per scene, and no language representation is attached to the Gaussians before a query.

At one level, the distinction from pre-query feature aggregation is

$$S_t^{\text{feat}}(g) = r(\mathcal{L}(F)_g, t), \qquad \ell_t(g) = \left[\Phi_{A,n}(\mathcal{L}(h_t(F)))\right]_g. \tag{2}$$

Here F stacks mask features, r evaluates feature relevance, h_t computes competitive query responses, and $\Phi_{A,n}$ denotes the regional averaging and statistical inference below. Feature aggregation combines whole-object, partial, and contextual observations before query evaluation, so the query cannot reweight them. Our response evaluation instead takes place at query time, before aggregation, and compares each mask with the other masks of its view, which aggregated features no longer allow, so the query weighs the original observations.

3.2 Competitive Query Responses

A response should reflect both a mask's agreement with the query and its standing among alternative regions in the same view (Fig. 1(a)). Relative competition alone would favor a region even when all candidates provide weak support. We therefore combine within-view competition with a null reference. Suppressing the level index, let $\mathcal{N}$ contain the normalized embeddings of "object," "things," "stuff," and "texture." We define

$$d_m = f_m^\top t - \max_{b\in\mathcal{N}} f_m^\top b, \qquad y_m = \frac{\exp(\beta d_m)}{M_v + \exp(\beta d_v^*)}, \tag{3}$$

where $d_v^* = \max_{m\in\mathcal{M}_v} d_m$ and $M_v = |\mathcal{M}_v|$. The margin is proportional to the logit of LERF's positive-versus-canonical relevance (Kerr et al., 2023). The response decomposes as $y_m = \exp(\beta(d_m - d_v^*))\sigma(\beta d_v^* - \log M_v)$: the first factor suppresses weaker relative responses, while the second attenuates observations with insufficient absolute support against the null reference of weight M_v.

3.3 Statistical Membership Inference

Accumulated responses do not carry a fixed meaning: the same response rate can indicate the target for one query and the background for another, and a query may have no target at all. We therefore interpret them with a model fitted separately for each query (Fig. 1(b)).

Observation Support Aggregation. Since each view responds mainly on its winning masks, we lift the responses to Gaussians and apply one mask-supported regional average, which shares them over the observed region, to obtain a response rate x_g (Appendix B.2). With n_g distinct views having

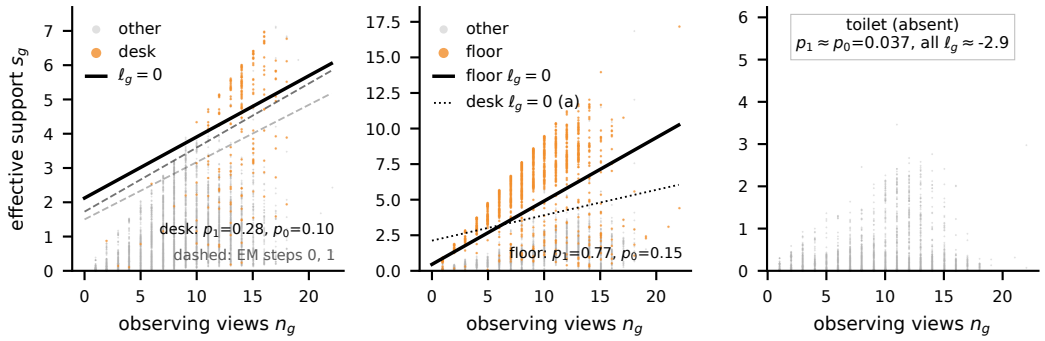

Figure 2: **Statistical membership inference with a per-query two-population mixture.** Each point is an observed Gaussian in one ScanNet scene, plotted by its observing-view count n_g and effective support $s_g = n_g x_g$ at the medium level; annotations give the fitted target and background response rates p_1 and p_0. (a) For *desk*, EM fits the two populations without labels, moving the boundary $\ell_g = 0$ from its initial estimates (dashed: steps 0 and 1) to the fitted boundary (solid). (b) Refitting for *floor* gives a much higher target rate, which tilts and shifts the boundary (dotted: *desk*). (c) For the no-target query *toilet*, the fitted rates nearly coincide, so every observed Gaussian receives only the prior log odds (-2.9) and the prediction is empty. Ground-truth labels color the points for display only.

a nonzero correspondence to g, the effective support $s_g = n_g x_g$ acts as a soft count of the views supporting g.

Two-Population Binomial Mixture. Each view acts as a noisy annotator of an observed Gaussian's target membership. Following the latent-class formulation of the Dawid–Skene model (Dawid & Skene, 1979), we assume that each Gaussian belongs to a target or background population, and that the views share one response rate per population: p_1 for the target and p_0 for the background, with $p_1 \geq p_0$. Since these annotations are soft and spatially coupled rather than independent binary annotations, we use a binomial working likelihood on the effective support. For latent membership $z_g \in \{0, 1\}$ and target proportion $\pi = P(z_g = 1)$, the accumulated support follows a two-population binomial mixture:

$$L_z(s_g, n_g) = p_z^{s_g}(1 - p_z)^{n_g - s_g}, \qquad P(s_g, n_g) = \pi L_1(s_g, n_g) + (1 - \pi) L_0(s_g, n_g). \tag{4}$$

Label-Free Fitting with EM. The parameters π, p_1, and p_0 are unknown and differ across queries. For each query and level, the expectation–maximization (EM) algorithm (Dempster et al., 1977) estimates them by maximum likelihood without labels. Starting from initial estimates that treat the highest response rates as a small target population, it alternates two steps: computing each Gaussian's posterior probability of target membership under the current parameters, and re-estimating p_1 and p_0 as the support rates of the soft target and background populations and π as the target share (Fig. 1(b); Appendix B.2). Gaussians with distinctly higher response rates thus form the target population, so the split comes from the query's own responses. The fitted parameters give the log posterior odds

$$\ell_g = \log \frac{\pi}{1 - \pi} + s_g \log \frac{p_1}{p_0} + (n_g - s_g) \log \frac{1 - p_1}{1 - p_0}. \tag{5}$$

Verdict and Decision Boundary. Eq. (5) adds a log-likelihood ratio for each observing view: its support adds $\log(p_1/p_0) > 0$, and its absence adds $\log \frac{1-p_1}{1-p_0} < 0$. The sign of ℓ_g is the verdict on membership (unlike an evaluation threshold; Appendix A); in the (n_g, s_g) plane of Fig. 2, the boundary $\ell_g = 0$ is a line whose slope depends only on p_1 and p_0 and whose position the prior shifts, so EM moves this line from its initial estimates (dashed) to the fitted boundary (solid; Figs. 1(b) and 2(a)), and refitting for each query moves it even within the same scene and views. In Fig. 2(b), the same response rate can place a Gaussian in the target population for *desk* and in the background for *floor*, and the boundary fitted for *floor* would select none of the desk; no fixed threshold on the response rate reproduces both boundaries. If the responses form no separate target group, the two populations collapse into one ($p_1 \approx p_0$), the likelihood ratios vanish, and only the prior log odds

remain, negative because π then stays near its small initial value (Appendix B.2); the prediction is then empty (Fig. 2(c)).

3.4 Gated Multi-Level Inference

No single mask granularity matches every target, so VerdictGS infers membership at every level and lets the query's evidence across levels, rather than predefined groups, determine the target's extent.

Gated Response Propagation. A target can appear as one mask at a coarse level and as several parts at a finer level, while a coarse mask can also merge distinct objects. Before each level's mixture is fitted, a region's response may support its constituent parts only if the region describes its parts semantically and the query favors the region over its best-matching part (Fig. 1(a), bottom). For a receiving mask m with a same-view source $C(m)$ that covers a fraction η_m of its association mass,

$$\widetilde{y}_m = \max\big(y_m, \eta_m y_{C(m)} \gamma_{C(m)} \sigma(\beta \Gamma_{C(m)})\big)\,, \tag{6}$$

where the semantic gate γ_C compares part-to-whole with inter-part similarity, and the query gate Γ_C averages the whole-versus-best-part margin difference across views (Appendix B.3).

Multi-Level Evidence Fusion. Each level then undergoes the inference of Sec. 3.3, and the log-odds fields bounded to $[-50, 50]$, $\bar{\ell}_g^k$, are summed (Fig. 1(c)):

$$e_g = \sum_{k \in \mathcal{K}} \bar{\ell}_g^k. \tag{7}$$

We treat e_g as an evidence score rather than joint posterior log odds, since levels share observations.

Membership-Conditioned Averaging. The summed evidence still varies within an object: a view that sees only a small part of an object often fails to recognize it, and this failure lowers only the Gaussians that view observes. Averaging within medium masks fills such parts, but an ordinary average also mixes target and background evidence in masks that extend beyond the target, trading recall for precision (Appendix G). Each medium mask therefore averages e_g separately over target and background Gaussians, weighted by the large-level membership $\omega_g = \sigma(\bar{\ell}_g^{\text{large}})$ and $1 - \omega_g$, into means μ^+ and μ^-, which each Gaussian blends by ω_g to produce S_t (Fig. 1(c); Appendix B.4); unobserved Gaussians take the score of their nearest observed neighbor.

4 Experiments

4.1 Experimental Setup

Benchmarks. We evaluate on LERF-OVS (Kerr et al., 2023; Qin et al., 2024) and ScanNet (Dai et al., 2017) with the LangSplat and OpenGaussian (Wu et al., 2024) evaluation code, respectively, and on 3D-OVS (Liu et al., 2023b) in Appendix C. On LERF-OVS (four scenes, 22 annotated frames), each query's field is rendered with the Gaussians' alpha-blending weights, and the code normalizes each map and applies a fixed threshold; we report mIoU. On ScanNet (ten scenes, Gaussians fixed at mesh vertices), each Gaussian takes the highest-scoring class of the official 19-class list or its 15- and 10-class subsets; we report mIoU and mAcc. These evaluation thresholds, not our decision boundary, set the reported extent; Sec. 4.4 evaluates the boundary against fixed thresholds. Since open-vocabulary queries may name absent objects but the benchmarks contain none, we propose adding no-target queries and reporting no-target and target accuracy (N-acc, T-acc), as in 2D generalized referring segmentation (Liu et al., 2023a), because N-acc alone would reward emptying every prediction (Appendix A). Appendix F evaluates VerdictGS under other works' 3D-selection thresholds.

Implementation. All scenes use frozen 2DGS geometry (Huang et al., 2024); the LERF-OVS scenes are those released by THGS (Dai et al., 2025). Masks and features follow the LangSplat preprocessing with SAM ViT-H (Kirillov et al., 2023) and OpenCLIP ViT-B/16 (Cherti et al., 2023). Each query is computed from its own text embedding and four fixed generic words, independently of other queries. We set $\beta = 100$, bound per-level log odds to ± 50, and fit each mixture with up to 60 EM iterations on at most 10^5 sampled Gaussians. Query inference runs on a GPU or a CPU with the same mIoU (Sec. 4.3). Further details are in Appendix B.

Table 1: **Segmentation on LERF-OVS.** Per-scene and mean mIoU (%); best in bold, second best underlined. Baselines are grouped by what they build before the query: per-scene optimization, per-Gaussian features, and object groups. †Evaluated with the method's own evaluation threshold instead of the LangSplat evaluation code (Appendix A).

Method	figurines	ramen	teatime	waldo	Mean ↑
LangSplatV2 (Li et al., 2025)	56.4	51.8	72.2	59.1	59.9
VALA (Wang et al., 2026)	59.9	51.5	70.2	<u>65.1</u>	61.7
Splat Feature Solver (Xiong et al., 2026)†	<u>67.6</u>	**62.3**	68.5	62.1	<u>65.1</u>
THGS (Dai et al., 2025)†	57.3	43.5	68.3	50.7	54.9
ReLaGS (Xie et al., 2026)†	64.7	51.2	**81.0**	60.6	64.4
VerdictGS (ours)	**67.8**	<u>60.1</u>	<u>73.6</u>	**72.3**	**68.5**

Table 2: **Segmentation on ScanNet.** mIoU and mAcc (%) for the official 19-, 15-, and 10-class lists; baselines are grouped as in Table 1. *Reported by NormLift (Zang et al., 2026).

	19 classes		15 classes		10 classes	
Method	mIoU ↑	mAcc ↑	mIoU ↑	mAcc ↑	mIoU ↑	mAcc ↑
LaGa (Cen et al., 2025b)	32.5	49.1	35.5	53.5	42.6	63.2
Splat Feature Solver (Xiong et al., 2026)*	33.3	51.4	36.4	55.4	44.7	63.5
THGS (Dai et al., 2025)	34.4	50.7	39.6	57.1	46.4	64.7
PairGS (Cha et al., 2026)	40.7	54.3	<u>44.8</u>	60.2	<u>53.0</u>	68.1
ExtrinSplat (Ding et al., 2026)	**45.5**	**58.4**	**47.2**	**61.7**	**53.7**	**74.9**
VerdictGS (ours)	<u>42.5</u>	<u>57.4</u>	43.8	<u>60.3</u>	50.9	<u>69.4</u>

4.2 Main Results

Quantitative Results. Without building language features or object groups before the query, VerdictGS achieves the highest mIoU on LERF-OVS, above per-scene optimization and both training-free designs: aggregating per-Gaussian features, which mixes observations before the query, and selecting among predefined object groups, which constrains the target's extent (Table 1). On ScanNet, it achieves the highest 19-class mIoU and mAcc among methods without an MLLM (Table 2) and remains competitive with the two methods that score higher, both of which build object groups before the query: ExtrinSplat also describes its groups with an MLLM and selects thresholds on LERF-OVS annotations, and PairGS leads in mIoU only with the shorter 15- and 10-class lists, where VerdictGS keeps a higher mAcc. On 3D-OVS, VerdictGS reaches 95.3 mIoU under the original full-class-list protocol (Appendix C).

Qualitative Results. Scoring each Gaussian once the query is given, rather than selecting among predefined groups, lets VerdictGS cover the whole target. THGS returns predefined superpoints that cover only part of the target (Fig. 3(a)) and returns them even for absent objects (Fig. 3(b)), whereas VerdictGS combines responses at every mask granularity and forms no target population for an absent object (see Appendix I for more results on LERF-OVS and ScanNet).

4.3 Preparation and Inference Cost

Table 3 compares what training-free methods build before the query and what each query then costs. Every baseline attaches a language representation to the Gaussians, such as per-Gaussian features, compressed codes, or a superpoint hierarchy, and builds it with a GPU. VerdictGS attaches none: it renders the mask–Gaussian contributions of each view once and accumulates them into query-independent correspondences and statistics, in less time than any baseline (Appendix B.5). Inferring at query time keeps each query light: it needs only 14 sparse matrix traversals and four three-parameter mixture fits. A query takes 16 ms on a GPU, the same order as most baselines (6–11 ms), and 0.49 s on one CPU thread, still faster than Dr. Splat, whose code search runs on the CPU; both devices give the same mIoU (Appendix H). Thus, although VerdictGS infers each query from multi-

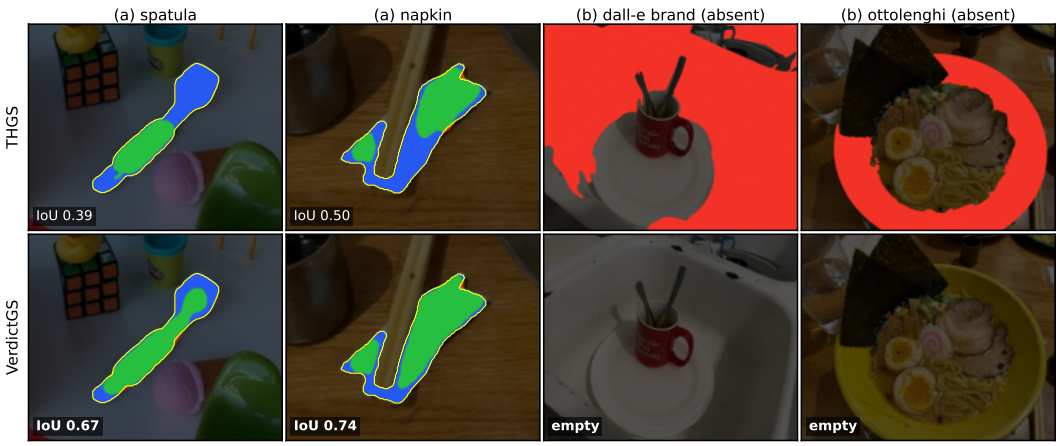

Figure 3: **Comparison with THGS on LERF-OVS.** (a) Target queries under the benchmark evaluation, cropped around the target (yellow: ground truth; green: correct; blue: missed; red: outside the target). (b) No-target queries rendered from the 3D selection: THGS returns its top three superpoints, whereas VerdictGS returns an empty prediction. Examples follow a fixed selection rule (See Appendix I for more detail).

view evidence, it prepares a scene faster than every baseline and answers a query in comparable time.

Table 3: **Pre-query construction and inference cost.** Measured on one machine (RTX 6000 Ada, EPYC 7313). Preparation is the wall-clock time from the same cached SAM and CLIP features and geometry; for VerdictGS, CPU cache construction plus GPU rendering of its contribution records. Query time covers one text query, from CLIP encoding to per-Gaussian results; Dr. Splat searches its codes on the CPU. VerdictGS gives the same mIoU on GPU and CPU (one thread; eight in parentheses).

Method	Language representation on Gaussians	Preparation	Query
Occam's LGS	Per-Gaussian features	100 s	10 ms
VALA	Per-Gaussian features	61 s	5.8 ms
Dr. Splat	Per-Gaussian PQ codes	214 s	595 ms
Splat Feature Solver	Per-Gaussian features	547 s	6.7 ms
THGS	Superpoint hierarchy	181 s	11 ms
VerdictGS (ours), GPU	**None** (mask–Gaussian correspondences)	38 s + 8 s	16 ms
VerdictGS (ours), CPU	**None** (mask–Gaussian correspondences)	38 s + 8 s	0.49 s (0.21 s)

4.4 Ablation and Analysis

Table 4 adds the stages of VerdictGS one at a time, separating when the query is evaluated, how membership is decided, and how the target's extent is completed.

Response Accumulation. Evaluating the query in each view before aggregation outperforms feature aggregation on every benchmark and in every LERF-OVS and ScanNet scene (Table 4). Aggregated features spread relevance beyond the target on LERF-OVS and miss parts of it on ScanNet, because aggregation mixes the target with its context and removes the comparison with the other regions of each view, so objects resembling the query also score high (see Appendix D for precision, recall, per-scene results, and examples).

Label-Free Decision Boundary. Fig. 2 shows in one scene how EM refits the boundary to each query without labels; over all queries, fitting the mixture instead of keeping its initial estimates raises mIoU on every benchmark (see Appendix E for what EM fits across queries and how it converges). Because the best fixed threshold differs across queries, a single threshold is a compromise that must be tuned with ground truth, and the best compromise differs between benchmarks; without tuning,

Table 4: **VerdictGS by stage.** Segmentation mIoU, N-acc, and T-acc (%; Sec. 4.1) on (a) LERF-OVS and (b) ScanNet (19 classes) under identical inputs. Rows follow Eq. (2) and Secs. 3.2, 3.3, and 3.4; the first three use one mask level and the same spatial averaging. No-target queries are other scenes' query names on LERF-OVS and absent classes on ScanNet; a Gaussian is selected when its relevance exceeds 0.5 for feature aggregation or its score exceeds zero otherwise (Appendix A).

(a) LERF-OVS

Stage	mIoU	N-acc	T-acc
Feature aggregation	50.1	54.1	88.1
Response accumulation	59.8	0.0	100.0
+ Binomial mixture	64.6	41.6	100.0
+ Multi-level inference	**68.5**	35.1	100.0

(b) ScanNet

Stage	mIoU	N-acc	T-acc
Feature aggregation	22.7	27.7	96.9
Response accumulation	35.3	0.0	100.0
+ Binomial mixture	34.6	59.6	86.5
+ Multi-level inference	**42.5**	54.3	91.7

Table 5: **Label-free boundary versus fixed thresholds.** Decision rules on identical ScanNet support statistics (one mask level, before the final spatial averaging), with mIoU computed from each target query's selected Gaussians rather than from class assignment, so values differ from Table 4. Fixed thresholds on the response rate x_g are chosen with ground truth: $\tau = 0.15$ gives the best segmentation, and $\tau = 0.3$ an N-acc similar to ours. Appendix E reports all thresholds and LERF-OVS.

Decision rule	Labels	mIoU ↑	N-acc ↑
Per-query binomial mixture (ours)	none	**29.4**	56.4
Fixed threshold $\tau = 0.15$	GT-tuned	29.1	18.1
Fixed threshold $\tau = 0.3$	GT-tuned	24.9	48.9

the per-query boundary segments close to it on each benchmark (Table 5; Appendix Fig. 6(a)). Even under the evaluation thresholds, which do not use this boundary, the mixture raises segmentation: removing it from the full pipeline lowers mIoU on every benchmark, most on 3D-OVS, where each pixel takes the highest-scoring class and the mixture's log odds place all queries on a common scale (Appendices C and E). At a single level, it lowers the ScanNet mIoU only in scenes where it leaves *the floor* empty (Table 4(b); Appendix E). The same boundary returns an empty prediction when the responses form no separate target population, unlike accumulated responses (Table 4; Fig. 2(c)).

Multi-Level Inference. Letting the query decide the target's extent requires responses at several mask granularities, as no single level is always best. Fusing level evidence with gated response propagation and membership-conditioned averaging completes objects that finer levels split into parts, keeps distinct objects apart, and recovers target queries that the single-level mixture leaves empty, at a modest cost in N-acc (Table 4; details in Appendices G and I).

5 Conclusion

We presented VerdictGS, which shows that a query's target can be inferred from multi-view observations once the query is given, treating each view as a noisy annotator of target membership. Under identical inputs, accumulating per-view responses instead of features improves segmentation on every benchmark, and a per-query mixture fitted with EM turns them into membership verdicts whose label-free boundary segments close to the best ground-truth-selected threshold. Without pre-query language features or object groups, VerdictGS achieves strong segmentation accuracy under each benchmark's official evaluation, prepares scenes faster than training-free baselines, and answers queries in comparable time. Modeling view reliability and dependence, as in the full Dawid–Skene model, would extend the mixture. Since open-vocabulary queries may name absent objects, a natural next step for 3D benchmarks is to score each method's own decision, both its extent without evaluation thresholds and its empty predictions for absent targets, as generalized referring segmentation does in 2D.

Limitations. VerdictGS removes the decision threshold but not the evaluation threshold: benchmarks contain no absent targets and score every method, including VerdictGS, with their own thresholds (Appendix A), so the reported accuracy measures its relevance field, not its decisions, whose extent and empty predictions appear only in Table 5 and N-acc/T-acc. It also inherits CLIP's recognition errors, and its mixture treats views as equally reliable, independent annotators.

ETHICS STATEMENT

This work studies open-vocabulary 3D segmentation on the publicly released LERF-OVS, ScanNet, and 3D-OVS datasets, used under their original terms. It does not involve new data collection, human participants, or personal data.

REPRODUCIBILITY STATEMENT

Section 3 describes the method, and Section 4.1 specifies the benchmarks, evaluation protocols, inputs, and hyperparameters, which are shared by all benchmarks. Appendix A details the no-target protocol, Appendix B specifies the EM initialization, updates, sampling seed, propagation gates, and execution order, and Appendix H reports the timing scope. Benchmark results compared with other methods are computed with each benchmark's official evaluation code; the no-target and decision-rule analyses and the 3D-selection results follow the protocols stated with them (Appendices A, E, and F). We will release the code upon publication.

AI USE STATEMENT

All research decisions in this work were made by the authors, including the problem formulation, the design principles, the evaluation protocols, the choice of the method among the explored variants, and the claims. Under the authors' direction, large language model assistants were used as tools to implement and test method variants, give feedback on experiments, run analyses and help interpret their results, summarize related work, and draft, translate, and edit text. Every reported number was checked against the saved outputs of the evaluation and analysis code, every reference was checked against its original publication, and the authors take full responsibility for the content of this paper.

REFERENCES

Jiazhong Cen, Jiemin Fang, Chen Yang, Lingxi Xie, Xiaopeng Zhang, Wei Shen, and Qi Tian. Segment any 3D Gaussians. In *Proceedings of the AAAI Conference on Artificial Intelligence (AAAI)*, 2025a. URL https://arxiv.org/abs/2312.00860.

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)*, 2025b. URL https://arxiv.org/abs/2505.24746.

Eunsung Cha, Hyunjoon Lee, and Jaesik Park. Relation-centric open-vocabulary 3D Gaussian segmentation. *arXiv preprint arXiv:2607.01140*, 2026. URL https://arxiv.org/abs/2607.01140.

Jiahuan Cheng, Jan-Nico Zaech, Luc Van Gool, and Danda Pani Paudel. Occam's LGS: An efficient approach for language gaussian splatting. In *British Machine Vision Conference (BMVC)*. BMVA, 2025. URL https://arxiv.org/abs/2412.01807.

Mehdi Cherti, Romain Beaumont, Ross Wightman, Mitchell Wortsman, Gabriel Ilharco, Cade Gordon, Christoph Schuhmann, Ludwig Schmidt, and Jenia Jitsev. Reproducible scaling laws for contrastive language-image learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 2818–2829, 2023.

Angela Dai, Angel X. Chang, Manolis Savva, Maciej Halber, Thomas Funkhouser, and Matthias Nießner. ScanNet: Richly-annotated 3D reconstructions of indoor scenes. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 5828–5839, 2017.

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, 2025. doi: 10.1145/3746027.3755146. URL https://arxiv.org/abs/2504.13153.

A. P. Dawid and A. M. Skene. Maximum likelihood estimation of observer error-rates using the EM algorithm. *Journal of the Royal Statistical Society. Series C (Applied Statistics)*, 28(1):20–28, 1979. doi: 10.2307/2346806.

Arthur P. Dempster, Nan M. Laird, and Donald B. Rubin. Maximum likelihood from incomplete data via the EM algorithm. *Journal of the Royal Statistical Society: Series B (Methodological)*, 39(1):1–38, 1977.

Jiayu Ding, Xinpeng Liu, Zhiyi Pan, Shiqiang Long, and Ge Li. ExtrinSplat: Decoupling geometry and semantics for open-vocabulary understanding in 3D Gaussian Splatting. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 31019–31028, 2026. URL https://arxiv.org/abs/2509.22225.

Binbin Huang, Zehao Yu, Anpei Chen, Andreas Geiger, and Shenghua Gao. 2D Gaussian Splatting for geometrically accurate radiance fields. In *SIGGRAPH 2024 Conference Papers*. Association for Computing Machinery, 2024. doi: 10.1145/3641519.3657428. URL https://surfsplatting.github.io/.

Minchao Jiang, Shunyu Jia, Jiaming Gu, Xiaoyuan Lu, Guangming Zhu, Anqi Dong, and Liang Zhang. VoteSplat: Hough voting Gaussian splatting for 3D scene understanding. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, 2025. URL https://arxiv.org/abs/2506.22799.

Kim Jun-Seong, GeonU Kim, Kim Yu-Ji, Yu-Chiang Frank Wang, Jaesung Choe, and Tae-Hyun Oh. Dr. Splat: Directly referring 3D Gaussian Splatting via direct language embedding registration. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 14137–14146, 2025. URL https://arxiv.org/abs/2502.16652.

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. 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 *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, 2023. URL https://www.lerf.io/.

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 *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, 2023. URL https://arxiv.org/abs/2304.02643.

LangSplat authors. Modified SAM automatic mask generator. Source code, n.d. URL https://github.com/minghanqin/segment-anything-langsplat/blob/main/segment_anything/automatic_mask_generator.py. Accessed September 24, 2026.

LERF authors. OpenCLIP encoder and canonical relevance implementation. Source code, n.d. URL https://github.com/kerrj/lerf/blob/main/lerf/encoders/openclip_encoder.py. Accessed September 24, 2026.

Wanhua Li, Yujie Zhao, Minghan Qin, Yang Liu, Yuanhao Cai, Chuang Gan, and Hanspeter Pfister. LangSplatV2: High-dimensional 3D language Gaussian splatting with 450+ FPS. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2025. URL https://arxiv.org/abs/2507.07136.

Chang Liu, Henghui Ding, and Xudong Jiang. GRES: Generalized referring expression segmentation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 23592–23601, 2023a. URL https://openaccess.thecvf.com/content/CVPR2023/html/Liu_GRES_Generalized_Referring_Expression_Segmentation_CVPR_2023_paper.html.

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)*, 2023b.

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. URL https://arxiv.org/abs/2503.22204.

Yuning Peng, Haiping Wang, Yuan Liu, Yipeng Lu, Zhen Dong, and Bisheng Yang. LEGO: Leveled language Gaussian splatting. In *European Conference on Computer Vision (ECCV)*, 2026. URL https://arxiv.org/abs/2608.10057.

Minghan Qin, Wanhua Li, Jiawei Zhou, Haoqian Wang, and Hanspeter Pfister. LangSplat: 3D language gaussian splatting. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2024. URL https://langsplat.github.io/.

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, 2021. URL https://proceedings.mlr.press/v139/radford21a.html.

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 *International Conference on 3D Vision (3DV)*, pp. 1791–1800, 2026. doi: 10.1109/3DV69130.2026.00170. URL https://arxiv.org/abs/2509.05515.

Simon K. Warfield, Kelly H. Zou, and William M. Wells. Simultaneous truth and performance level estimation (STAPLE): An algorithm for the validation of image segmentation. *IEEE Transactions on Medical Imaging*, 23(7):903–921, 2004.

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.

Yaxu Xie, Abdalla Arafa, Alireza Javanmardi, Christen Millerdurai, Jia Cheng Hu, Shaoxiang Wang, Alain Pagani, and Didier Stricker. ReLaGS: Relational language gaussian splatting. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 23826–23836, 2026. URL https://arxiv.org/abs/2603.17605.

Butian Xiong, Rong Liu, Kenneth Xu, Meida Chen, and Andrew Feng. Splat Feature Solver. In *International Conference on Learning Representations (ICLR)*, 2026. URL https://openreview.net/forum?id=AepuXqQM4X.

Yihan Zang, Da Li, Dominik Engel, Shinkyu Park, and Ivan Viola. NormLift: From lifted features to semantic reliability in 3D Gaussian Splatting. *arXiv preprint arXiv:2609.18898*, 2026. URL https://arxiv.org/abs/2609.18898.

A Evaluation Protocols

Two Kinds of Thresholds. Segmenting a query's target from relevance scores involves two thresholds. A *decision threshold* belongs to the method and decides which Gaussians or regions form the target; an *evaluation threshold* belongs to the evaluation and turns each method's output into the masks or labels scored against the annotations. Existing methods set a decision threshold on their relevance scores before the query: a value selected on the evaluation annotations (Jun-Seong et al., 2025; Ding et al., 2026; Zang et al., 2026), a rule that changes across benchmarks (Cheng et al., 2025), a cut on scores normalized within each query (Wang et al., 2026), or a preset fraction of the maximum score or number of top-scoring groups (Peng et al., 2026; Dai et al., 2025; Lu et al., 2025). VerdictGS sets none: its label-free decision boundary, the sign of per-query log posterior odds whose parameters EM fits without labels, amounts to a threshold on the response rate fitted to each query (Appendix B.2). Table 5 and Fig. 2 evaluate this decision. The same decision also answers whether the target is present: a query whose fused evidence is positive at no observed Gaussian receives an empty prediction, which N-acc and T-acc evaluate (below), whereas the extent reported on the benchmarks is set by their evaluation thresholds.

Evaluation Thresholds across Papers. Evaluation thresholds also differ across papers. The LERF-OVS code normalizes each rendered map within its frame and applies a fixed threshold, and the ScanNet and 3D-OVS codes assign each point or pixel its highest-scoring class; several methods instead report LERF-OVS results with their own selection and binarization rules, which serve as both their decision and evaluation thresholds, are designed for their output, and are sometimes set with the annotations, for instance from the number of annotated objects (Xiong et al., 2026). We apply each benchmark's official evaluation code unchanged to the relevance field of VerdictGS, so our reported mIoU involves neither our decision nor any threshold that we selected, and Table 1 marks the methods evaluated with their own rules.

No-Target Evaluation. Open-vocabulary queries are posed without knowing what a scene contains, so a query may name an absent object, yet the benchmarks contain no such queries; we therefore add no-target queries and report N-acc and T-acc, the rates of empty predictions on no-target queries and nonempty predictions on target queries, as generalized referring segmentation does in 2D (Liu et al., 2023a). For LERF-OVS and 3D-OVS, no-target queries are the query names of other scenes in the same benchmark that are absent from the current scene's query list; for ScanNet, they are classes absent from its labels. A prediction is empty when no observed Gaussian is selected before benchmark evaluation. Because absence in LERF-OVS and 3D-OVS is determined by name, a similar object may still exist, making N-acc a conservative estimate.

B Additional Method Details

B.1 Scene Geometry and Observation Caches

We use frozen 2DGS geometry (Huang et al., 2024), whose primitives are Gaussian surfels embedded in 3D space. Inference uses rendering-contribution correspondences without requiring the primitives' shape parameters. The LERF-OVS scenes are the 2DGS reconstructions released by THGS (Dai et al., 2025), and the 3D-OVS scenes are reconstructed with 2DGS using the THGS implementation. For ScanNet, Gaussians are initialized at the mesh vertices and trained with fixed positions and without densification or pruning, so that each Gaussian corresponds to one vertex, as the OpenGaussian evaluation requires (Wu et al., 2024).

We use cached masks and features in the LangSplat preprocessing format (Qin et al., 2024). Its modified SAM generator (LangSplat authors, n.d.) separates the three multimask outputs into small, medium, and large branches, while its default branch pools these outputs before filtering and non-maximum suppression. Consequently, default is a mixed-granularity proposal set. It is not an additional mask size above medium or large.

Observation–Gaussian Association. The diagonal normalizers and observing-view count are

$$D_G^k = \mathrm{diag}(A^k \mathbf{1}), \qquad D_M^k = \mathrm{diag}((A^k)^\top \mathbf{1}),$$
$$n_g^k = \sum_{v \in \mathcal{V}} \mathbf{1}\left[\sum_{m \in \mathcal{M}_v^k} A_{gm}^k > 0\right]. \tag{B1}$$

The count measures correspondence coverage, not geometric visibility established independently of the masks. The scene preparation retains only masks with positive total association mass. Features are normalized before query comparisons, and all zero-denominator averages return zero.

B.2 Statistical Membership Inference

Observation Support Aggregation. Let $\widetilde{y}$ denote responses after propagation, retaining the original responses at levels without incoming transfer. At each level, we form

$$x = \mathcal{L}(\mathcal{R}(\mathcal{L}(\widetilde{y}))), \qquad s_g = n_g x_g. \tag{B2}$$

The inner lift forms a Gaussian response field. Averaging it over mask support and lifting it back shares responses among Gaussians associated with the same observed regions. This averaging precedes mixture fitting and therefore changes the support statistics used by the model. The observing-view count remains the correspondence count in Eq. (B1).

From View Annotations to a Count Model. Consider an idealized binary annotation R_{vg} for each view that observes Gaussian g. Let z_g be its latent target membership and suppose $P(R_{vg} = 1 \mid z_g = z) = p_z$, shared across views. Under conditional independence, the likelihood of these annotations depends on their positive count K_g and observing-view count n_g through

$$\prod_v P(R_{vg} \mid z_g = z) = p_z^{K_g}(1 - p_z)^{n_g - K_g}. \tag{B3}$$

This is the binomial likelihood kernel. The connection to Dawid–Skene (Dawid & Skene, 1979) is the joint estimation of latent membership and class-conditional response tendencies from unlabeled annotations. Here the response tendencies are shared across views. The two rates remain distinct: p_1 is sensitivity and p_0 is the false-positive rate in the idealized binary interpretation.

Actual mask responses are continuous, rendering-weighted, and spatially averaged. We therefore substitute effective support $s_g = n_g x_g$ for the ideal count K_g; s_g is not an observed count of independent binary annotations. The posterior quantities describe the fitted response populations, and their calibration against ground-truth membership is not assumed.

Query-Dependent Decision Boundary. For $p_1 > p_0$ and $n_g > 0$, the sign of the single-level log odds is equivalent to

$$\ell_g > 0 \quad \Longleftrightarrow \quad x_g > \tau_t(n_g), \qquad \tau_t(n) = \frac{-n^{-1} \log \frac{\pi}{1-\pi} - \log \frac{1-p_1}{1-p_0}}{\log \frac{p_1(1-p_0)}{p_0(1-p_1)}}. \tag{B4}$$

The threshold is determined by the fitted query parameters and observing-view count. It need not lie in $[0, 1]$, allowing the rule to select all or none of the observed Gaussians in limiting cases. If $p_1 = p_0$, the likelihood ratio vanishes and the log odds reduce to the prior odds; since the likelihood then no longer depends on π, EM leaves π near its initial value $\pi^{(0)} = 0.05$, which treats the target as a minority of the observed Gaussians and keeps these log odds negative. A prediction is empty when no observed Gaussian has positive evidence.

Sampling and Initialization. At each level, we select at most 10^5 indices uniformly without replacement from Gaussians with $n_g > 0$, using NumPy's default random generator with seed 20260920. Selected indices are sorted and reused across queries. The response rates are clipped to $[0, 1]$ before fitting. On this sample $\mathcal{I}$, we initialize

$$\pi^{(0)} = 0.05, \qquad p_1^{(0)} = \max(Q_{0.95}(x), 10^{-3}), \qquad p_0^{(0)} = \max(\text{median}(x), \epsilon), \quad \epsilon = 10^{-4}. \tag{B5}$$

EM Updates. At each iteration, clamp the response rates to $[\epsilon, 1-\epsilon]$, compute the log odds from the current parameters, and obtain responsibilities $r_g = \sigma(\ell_g)$. The updates are

$$\begin{aligned} \pi^+ &= \text{clip}\left(\frac{1}{|\mathcal{I}|}\sum_{g\in\mathcal{I}} r_g, \epsilon, 1-\epsilon\right), \\ \widehat{p}_1 &= \frac{\sum_{g\in\mathcal{I}} r_g s_g}{\max(\sum_{g\in\mathcal{I}} r_g n_g, 10^{-9})}, \\ \widehat{p}_0 &= \frac{\sum_{g\in\mathcal{I}} (1-r_g) s_g}{\max(\sum_{g\in\mathcal{I}} (1-r_g) n_g, 10^{-9})}, \\ p_1^+ &= \max(\widehat{p}_1, \widehat{p}_0), \qquad p_0^+ = \min(\widehat{p}_1, \widehat{p}_0). \end{aligned} \tag{B6}$$

The ordered rates are used in the next iteration. EM runs for at most 60 iterations and stops earlier if all three parameters stop changing. Before final scoring, all three parameters are clamped to $[\epsilon, 1-\epsilon]$. An empty fitting sample produces zero evidence. Otherwise, the fitted parameters are applied to every Gaussian at that level, with $\ell_g = 0$ for $n_g = 0$.

Relation to LERF Relevance. For normalized features and the same negative embeddings, LERF's canonical comparison is $r_{\text{LERF}} = \min_b \sigma(\kappa(f_m^\top t - f_m^\top b)) = \sigma(\kappa d_m)$. Hence $\text{logit}(r_{\text{LERF}}) = \kappa d_m$. The official implementation uses the temperature $\kappa = 10$ (LERF authors, n.d.). VerdictGS uses this contrastive margin within a different response rule, including competition between masks and a null reference.

B.3 Gated Response Propagation

Because responses compete only within one level, a finer level gives little support to a whole object that it splits into weakly matching parts, so propagation lets the coarse region's response support these parts, and the gates keep it from spreading onto distinct objects that a coarse mask merges. Most predictions change little, so the mean gain is small, but some objects that a finer level splits into parts are recovered (Table 18).

For a receiving mask m at level k and a candidate C at level k' in the same view, define

$$\rho_{mC} = \frac{\sum_g A_{gm}^k \mathbf{1}[A_{gC}^{k'} > 0]}{\sum_g A_{gm}^k}, \qquad C(m) = \arg\max_C \rho_{mC}, \qquad \eta_m = \rho_{m,C(m)}. \tag{B7}$$

Only candidates with positive overlap are retained; unmatched receiving masks keep their original responses. This is an association in Gaussian support and does not assume a strictly nested mask hierarchy or a larger source-mask area.

Let $\mathcal{P}_C = \{m : C(m) = C\}$ and $w_m = \eta_m \sum_g A_{gm}^k$. The semantic consistency statistic is

$$\begin{aligned} a_C = &\frac{\sum_{m\in\mathcal{P}_C} w_m f_m^\top f_C}{\sum_{m\in\mathcal{P}_C} w_m} \\ &- \frac{\sum_{\substack{m,j\in\mathcal{P}_C \\ m\neq j}} w_m w_j f_m^\top f_j}{\sum_{\substack{m,j\in\mathcal{P}_C \\ m\neq j}} w_m w_j}, \qquad \gamma_C = \sigma(\beta a_C). \end{aligned} \tag{B8}$$

The first term asks whether the source feature provides a common description of its assigned parts; the second supplies a reference based on the parts' own semantic agreement. This statistic is a cue

for semantic unity, not a guarantee that the region contains exactly one physical object. With fewer than two assigned parts, $\gamma_C = 1$; implementations cap the sigmoid's upper argument at 50.

The query gate compares the source margin with its best-matching assigned part and averages this difference across views using source responses as weights:

$$\delta_C = d_C - \max_{m \in \mathcal{P}_C} d_m, \qquad \Gamma = \mathcal{R}_{k'}\left(\frac{A^{k'}(y^{k'} \odot \delta)}{A^{k'} y^{k'}}\right). \tag{B9}$$

We set $\delta_C = 0$ for an empty part set. All margins come from the original feature comparisons, while $y^{k'}$ denotes the current source responses. The semantic and query gates use the same $\beta = 100$ as response competition.

Large and default each provide a candidate update to medium according to Eq. (6); the two updates are combined by a maximum. The updated medium responses then propagate to small. Thus, medium-to-small propagation also uses the updated medium responses as weights in Eq. (B9). The maximum in Eq. (6) guarantees $\widetilde{y}_m \geq y_m$, although subsequent mixture refitting need not preserve the ordering of final evidence scores.

B.4 Multi-Level Evidence Fusion and Membership-Conditioned Averaging

Multi-Level Evidence Fusion. We bound each level's log odds before summation and derive the guide from the large level:

$$\bar{\ell}_g^k = \mathrm{clip}(\ell_g^k, -\lambda, \lambda), \qquad \omega_g = \sigma(\bar{\ell}_g^{\mathrm{large}}), \qquad \lambda = 50. \tag{B10}$$

Bounding controls extreme contributions during fusion; exponential overflow is handled separately in the E-step. The sum in Eq. (7) is an evidence score rather than an exact joint posterior log odds: levels share observations, and each fitted score includes its own prior term. The large-level guide supplies broad region context for conditional averaging on medium masks.

Membership-Conditioned Averaging. For each medium mask, we estimate two means weighted by ω and $1 - \omega$, and combine their lifted values according to each Gaussian's guide:

$$\begin{aligned} \mu^+ &= \frac{\mathcal{R}_{\mathrm{medium}}(\omega \odot e)}{\mathcal{R}_{\mathrm{medium}}(\omega)}, \\ \mu^- &= \frac{\mathcal{R}_{\mathrm{medium}}((\mathbf{1} - \omega) \odot e)}{\mathcal{R}_{\mathrm{medium}}(\mathbf{1} - \omega)}, \\ S_t &= \mathcal{L}_{\mathrm{medium}}(\mu^-) + \omega \odot \mathcal{L}_{\mathrm{medium}}(\mu^+ - \mu^-). \end{aligned} \tag{B11}$$

The final field is a relevance score rather than a calibrated posterior, but its zero is not tuned: each level adds bounded log odds that are negative wherever its evidence favors the background, and the conditional averaging takes weighted means with nonnegative weights, so a query whose fused evidence is negative at every observed Gaussian receives an empty prediction. The equal-posterior interpretation of zero holds exactly only for the individual mixture log odds.

Nearest-Neighbor Completion. Completion uses the set $\mathcal{O} = \{g : n_g^{\mathrm{medium}} > 0\}$. For $g \notin \mathcal{O}$, find $j^*(g) = \arg\min_{j \in \mathcal{O}} \|\mathbf{x}_g - \mathbf{x}_j\|_2$ and copy $S_t(j^*(g))$. The map is built once per scene and reused for all queries. If no observed source exists, unsupported values remain zero.

B.5 Execution Details

The execution order is: compute margins and responses at all four levels; propagate large and default responses to medium; propagate the updated medium responses to small; compute each level's regional response and observation support; fit four query-specific mixtures and evaluate bounded log odds; sum the scores and perform conditional averaging; complete unobserved Gaussians.

The CPU implementation groups multiple right-hand sides at matching dependency stages, so each query requires 14 sparse matrix traversals. It caches row and column normalizers, observing-view counts, generic-negative scores, cross-level associations, semantic gates, fitting indices, and completion indices. The four mixture fits estimate three scalar parameters each. Query batches share matrix access but do not pool response statistics or fitted parameters across queries.

C 3D-OVS RESULTS

Protocol. We use the five 3D-OVS (Liu et al., 2023b) scenes evaluated by LangSplat (Qin et al., 2024) (bed, bench, lawn, room, and sofa), which contain long-tail objects and annotated views. Following the original protocol, the field of each class in the scene's class list is rendered to the annotated views, each pixel takes the class with the highest score, and mIoU is averaged within each scene and then over the five scenes. VerdictGS computes each class's field from its own text embedding and the four generic words, independently of the other classes, so only the evaluation code uses the class list, in this final assignment. Later methods also report a per-class protocol in which each class is segmented separately; since the two protocols differ in whether classes compete at each pixel, Table 7 lists them separately.

Stage Analysis. The per-query mixture contributes more on 3D-OVS than any other stage (Table 6). Because each pixel takes the highest-scoring class, the fields of different queries are compared directly: response rates are not comparable across queries, whose distributions differ (Appendix E), whereas the log posterior odds of each query's mixture share a common zero at equal posterior. Beyond this common zero, the log odds also count evidence against each query: where a query's mixture has a target population, every view that observes a Gaussian without supporting the query lowers its score (Eq. 5). Each pixel therefore goes to the class that its views support most and contradict least, so even *white sheet*, which forms no target population, takes its pixels: on them, it scores about -10 and every other class -80 or below. As every row of Table 6 uses the same class list and evaluation, this gain comes from the mixture rather than from the closed class list. ScanNet uses the same kind of evaluation, but there the single-level mixture also leaves broad surfaces empty (Appendix E), which offsets its gain. The two target classes that VerdictGS leaves empty (T-acc 93.5) are *white sheet* and *grey sofa*, plain surfaces covering most of their scenes: at the medium and large levels, a mask matches the query better than the generic words in at most a quarter of the views, so no target population forms.

Table 6: VerdictGS by stage on 3D-OVS (%), with the rows, inputs, and N-acc/T-acc selection rule of Table 4; no-target queries are other scenes' class names (Appendix A).

Stage	mIoU	N-acc	T-acc
Feature aggregation	67.7	79.8	90.3
Response accumulation	71.9	0.0	100.0
+ Binomial mixture	89.7	82.3	90.3
+ Multi-level inference	**95.3**	79.8	93.5

Comparison. Under the original protocol, VerdictGS is second only to SAGA, which optimizes a feature field for each scene, and exceeds the per-scene 3D-OVS baseline (Table 7). Per-class results are listed for reference, since each method applies its own selection rule and averaging. With the LangSplat evaluation rule, which scores each class's field alone as on LERF-OVS, VerdictGS reaches 86.0, with 26 of the 31 classes above 0.8 IoU; the largest loss is *grey sofa* (0.0), which forms no target population as described above, whereas under the original protocol it still takes its pixels because the other classes score even lower there. This reveals a limitation of VerdictGS: it identifies a target only when the views support it distinctly more than the rest of the scene, so a target that no view singles out, such as a plain surface covering most of the scene, cannot be told apart from an absent one. Our future work aims to capture such weakly supported targets, for instance through the spatial coherence of their support or their surface geometry, while still returning empty predictions for absent targets.

Table 7: 3D-OVS results (mIoU, %), grouped by evaluation protocol. Baseline results are taken from the cited papers. †Five-scene mean computed by LangSplat (Qin et al., 2024).

Method	Training	mIoU ↑
Original protocol: each pixel or region takes the highest-scoring class of the scene		
3D-OVS (Liu et al., 2023b)†	Per-scene	86.8
SAGA (Cen et al., 2025a)	Per-scene	**96.0**
VerdictGS (ours)	Training-free	<u>95.3</u>
Per-class protocol: each class is segmented separately		
Splat Feature Solver (Xiong et al., 2026)	Training-free	86.3
LangSplat (Qin et al., 2024)	Per-scene	93.4
LangSplatV2 (Li et al., 2025)	Per-scene	94.6
THGS (Dai et al., 2025)	Training-free	94.9
Occam's LGS (Cheng et al., 2025)	Training-free	95.0
LaGa (Cen et al., 2025b)	Per-scene	95.3
LEGO (Peng et al., 2026)	Per-scene	96.5

D Response Accumulation versus Feature Aggregation

This section supports the response accumulation analysis of Sec. 4.4, detailing the first two rows of Table 4. Feature aggregation lifts the CLIP mask features of one level to each Gaussian with the rendering-weighted average $\mathcal{L}(F)$ of Eq. (1), the aggregation rule shared by training-free lifting methods (Cheng et al., 2025; Jun-Seong et al., 2025; Xiong et al., 2026) without their additional filtering or compression, and scores the lifted feature with LERF's relevance against the same generic words, $\sigma\big(10\,(f_g^\top t - \max_{b\in\mathcal{N}} f_g^\top b)\big)$ (Kerr et al., 2023). Response accumulation lifts the competitive responses of Eq. (3) from the same masks. Both share the correspondences, the completion of unobserved Gaussians, and the benchmark evaluation, and neither fits a mixture or propagates across levels.

Spatial Averaging. Spatial averaging raises response accumulation on all three benchmarks but lowers feature aggregation on LERF-OVS and ScanNet (Table 8). A response is high only on masks that win the query in their view, so lifted responses cover only the parts of the target seen as winning masks; averaging over mask regions shares them across the target and restores recall with little loss of precision. A lifted feature is already an average over observations, and averaging it again further blurs the target with its surroundings. Without averaging, response accumulation is therefore precise but incomplete, and it falls below feature aggregation on LERF-OVS; Table 4 applies the same averaging to both.

Table 8: Feature aggregation versus response accumulation under identical inputs (medium level; no mixture fitting or propagation). mIoU, recall (Rec.), and precision (Prec.) in %, computed from each benchmark's evaluated selections and averaged like mIoU, with precision over nonempty selections. Spatial averaging applies the regional average of Eq. (B2) and a final mask average. Choosing the feature relevance temperature and averaging on the evaluation labels narrows but does not close the gap on any benchmark, and applying LERF relevance to each mask before lifting, without comparing the masks of a view, performs like feature aggregation.

		LERF-OVS			ScanNet			3D-OVS
Lifted	Averaging	mIoU	Rec.	Prec.	mIoU	Rec.	Prec.	mIoU
Features	none	52.4	85.8	60.9	24.4	40.1	51.9	64.2
Features	spatial	50.1	90.5	54.3	22.7	38.5	61.0	67.7
Responses	none	42.1	43.3	80.5	30.5	47.2	51.0	70.4
Responses	spatial	**59.8**	63.6	79.0	**35.3**	52.4	60.9	**71.9**

Precision and Recall. Under the same averaging, response accumulation outperforms feature aggregation on all three benchmarks and in every LERF-OVS and ScanNet scene (Table 9), but for different reasons. On LERF-OVS the gain comes from precision (Fig. 4): once features are aggregated, a Gaussian's relevance can no longer be compared with the other regions of the views that observed it, so objects resembling the query also receive high relevance, whereas a response

is high only where a mask wins the query within its view. On ScanNet it comes from recall, consistent with mixed features favoring a class observed with the target when each Gaussian takes the highest-scoring class.

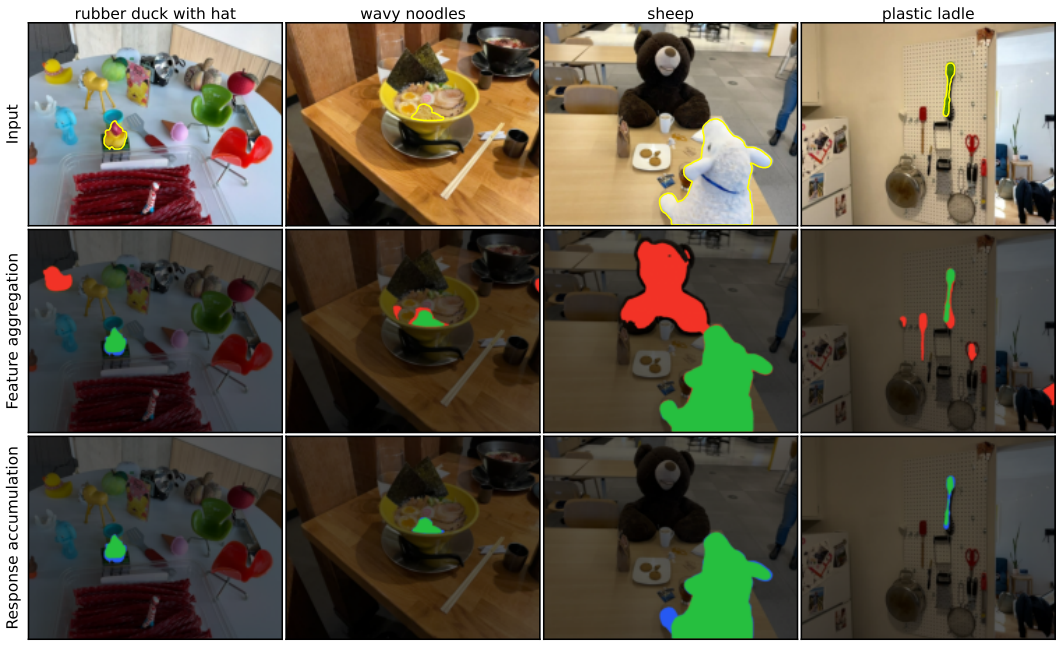

Figure 4: Feature aggregation versus response accumulation on LERF-OVS under identical inputs and spatial averaging (Table 8), shown with the benchmark evaluation (yellow: ground truth; green: correct; blue: missed; red: outside the target). Feature aggregation also selects objects resembling the query or next to it, such as another duck, the teddy bear, other utensils, and neighboring contents of the bowl; response accumulation keeps the object that wins in each view. In each scene, the example is the frame–query pair whose gain is closest to the mean gain among the pairs where response accumulation is better.

Table 9: Per-scene mIoU (%) of the stages in Table 4. ScanNet uses the 19-class list.

LERF-OVS	figurines	ramen	teatime	waldo	Mean
Feature aggregation	43.4	31.2	65.2	60.6	50.1
Response accumulation	53.4	51.6	68.3	65.8	59.8
+ Binomial mixture	58.6	56.6	73.0	70.2	64.6
+ Multi-level inference	67.8	60.1	73.6	72.3	68.5

ScanNet	0000	0062	0070	0097	0140	0200	0347	0400	0590	0645	Mean
Feature aggregation	23.0	39.3	18.6	22.3	16.2	22.7	19.4	17.9	24.8	22.6	22.7
Response accumulation	42.5	45.6	31.7	32.3	27.9	34.2	29.2	35.5	35.2	39.0	35.3
+ Binomial mixture	45.7	53.2	30.0	21.2	22.3	17.6	31.2	41.7	41.0	42.6	34.6
+ Multi-level inference	50.7	52.6	34.0	46.4	31.6	25.3	39.0	58.0	40.5	47.0	42.5

Using Several Levels. Using more mask levels does not let feature aggregation close the gap (Table 10). Whether it takes one level, averages relevance over the four levels, keeps the level with the highest relevance for each query, or lets the LangSplat evaluator choose a level for each frame, it stays below single-level response accumulation on all three benchmarks. Each option selects or averages relevance computed from features already mixed within a level, so none can reweight the individual observations for the query.

Table 10: Feature aggregation with different uses of the four mask levels (mIoU, %). The per-frame level choice of the LangSplat evaluator applies only to LERF-OVS. The last row is single-level response accumulation with spatial averaging, the second row of Table 4.

Configuration	LERF-OVS	ScanNet	3D-OVS
Features, default level	49.5	29.9	62.4
Features, small level	37.3	15.2	49.0
Features, medium level	52.4	24.4	64.2
Features, large level	49.9	32.6	63.5
Features, relevance averaged over the four levels	51.1	27.6	63.3
+ final mask average	50.1	27.8	68.5
Features, level with the highest relevance per query	50.4	29.9	59.3
Features, level chosen per frame by the LangSplat evaluator	52.6	–	–
Response accumulation (medium, spatial averaging)	**59.8**	**35.3**	**71.9**

E LABEL-FREE DECISION BOUNDARY

This section supports the analysis of the label-free decision boundary in Sec. 4.4, first extending Fig. 2 from one scene to all queries and then comparing other decision rules. Unless noted otherwise, the rules act on the same medium-level ScanNet support statistics as Table 5 (96 target and 94 no-target queries), without propagation, level fusion, or final averaging; target-query IoU is computed on labeled Gaussians and averaged over classes within each scene and then over scenes. Table 12 also applies the rules to the medium-level statistics of LERF-OVS (67 target and 185 no-target queries).

What EM Fits. For target queries, EM finds a target population whose response rate is several times that of the background (Table 11), and it moves the boundary from its initial estimates in a direction that depends on the query: upward for 59 of the 96 target queries, which then select fewer Gaussians, and downward for 34, which select more. The initial estimates assume that the target covers 5% of the observed Gaussians: every query whose boundary falls ends with a larger target proportion and a lower target response rate, as for *wall*, whereas the boundary rises mostly where the target proves smaller or its response rate higher, as for *toilet* (Fig. 5). For most no-target queries, the two rates coincide, and these queries account for nearly all empty predictions.

Table 11: Fitted mixtures on the ScanNet support statistics of Table 5: median ratio of the target to the background response rate, queries whose two rates coincide ($p_1/p_0 < 1.01$), and queries with an empty prediction.

Queries	Median p_1/p_0	$p_1/p_0 < 1.01$	Empty
Target (96)	6.4	10	11
No-target (94)	1.0	51	53

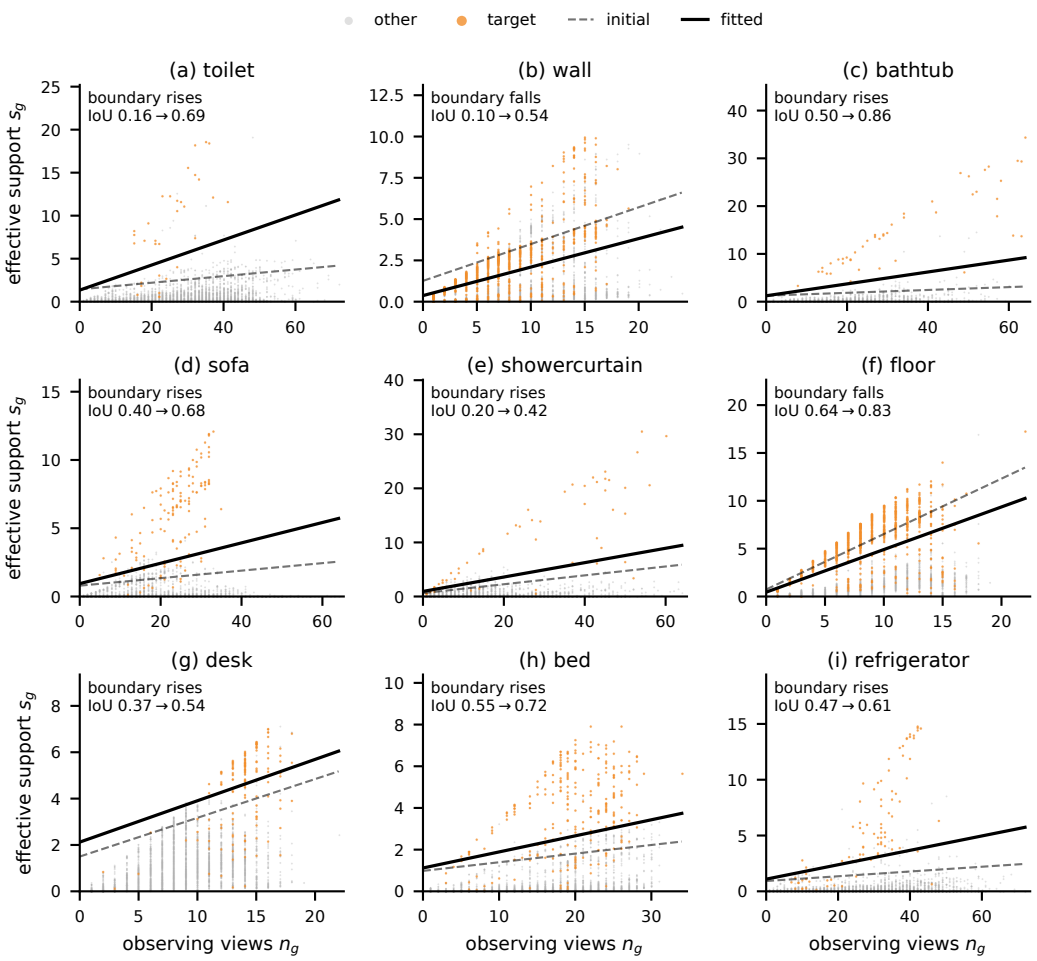

Figure 5: EM moves the boundary of each target query from its initial estimates (dashed) to the fitted boundary (solid) in a direction that depends on the query, on the ScanNet support statistics of Table 5. The boundary rises where the initial estimates also select background, as for *toilet* and *bathtub*, and falls where they miss much of the target, as for *wall* and *floor*. For each class, the example is the target query with the largest IoU gain, and the nine classes with the largest gains are shown in that order; ground-truth labels color the points for display only.

Fitting and Convergence. The accuracy comes from fitting each query, not only from the form of the model: with the initial estimates of Eq. (B5), the full method falls below the fitted result on every benchmark, most on 3D-OVS, and five iterations already come close to the final accuracy (Fig. 6(c)). The log-likelihood increases at every iteration for every query, and half of the queries come within 10^{-3} of their final parameters in 12 iterations.

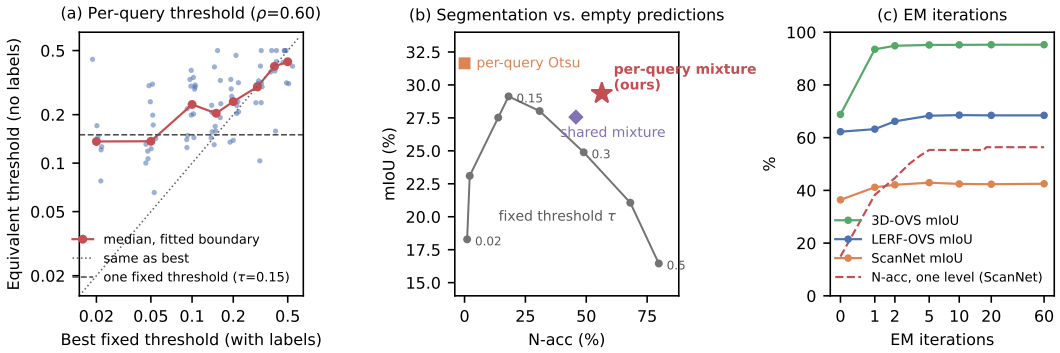

Figure 6: Label-free boundary on ScanNet. (a) For each target query, the fixed threshold with the highest IoU, chosen with labels, against the equivalent threshold of the fitted boundary, i.e., the fixed threshold that selects as many Gaussians (interpolated between the candidates; the 11 target queries left empty are omitted); a single fixed threshold is a horizontal line. (b) Segmentation versus N-acc for the fixed-threshold sweep, per-query Otsu, a mixture shared by the queries of a scene, and the per-query mixture. (c) Full-method mIoU on the three benchmarks and N-acc of the one-level mixture by the number of EM iterations; zero iterations uses the initial estimates.

The Best Threshold Varies across Queries. A single threshold cannot suit every query (Fig. 6(a)). For each target query, the fixed threshold with the highest IoU ranges over all eight candidates, and the threshold that is best overall is best for only a small share of the queries: it selects too much where a higher threshold is best and too little where a lower one is. The fitted boundary follows each query's best threshold without labels, so it gains over the single best fixed threshold on the queries whose best threshold lies far from it and trails it on those near it.

Decision Rules. On ScanNet, the per-query mixture is the only rule that matches the best fixed threshold in segmentation while returning empty predictions for most no-target queries, and it needs no labels (Fig. 6(b), Table 12). Fixed thresholds trade segmentation for N-acc along the sweep. Per-query Otsu thresholding adapts to each query and segments best among the label-free rules, but as a relative rule it always selects the higher-response class and never returns an empty prediction. A mixture shared by all queries of a scene is worse on both measures, so the model must be fitted per query.

Table 12: Decision rules on the support statistics of Table 5 and on the same one-level statistics of LERF-OVS (%). The per-class oracle chooses a threshold for each class on the labels and bounds target-query segmentation for threshold rules; it defines no rule for no-target queries. Dashes mark values that are undefined or, on LERF-OVS, not computed.

		ScanNet			LERF-OVS		
Decision rule	Labels	mIoU	N-acc	T-acc	mIoU	N-acc	T-acc
Per-query binomial mixture	none	29.4	56.4	88.5	57.3	41.6	100.0
Per-query Otsu	none	31.6	0.0	100.0	62.6	0.0	100.0
Shared mixture	none	27.6	45.7	95.8	—	—	—
Fixed $\tau = 0.02$	GT-tuned	18.3	1.1	100.0	33.1	17.3	100.0
Fixed $\tau = 0.05$	GT-tuned	23.1	2.1	100.0	50.7	29.2	100.0
Fixed $\tau = 0.1$	GT-tuned	27.5	13.8	100.0	60.8	40.5	98.5
Fixed $\tau = 0.15$	GT-tuned	29.1	18.1	99.0	58.6	51.4	97.0
Fixed $\tau = 0.2$	GT-tuned	28.0	30.9	97.9	56.0	60.5	89.6
Fixed $\tau = 0.3$	GT-tuned	24.9	48.9	92.7	50.2	75.7	83.6
Fixed $\tau = 0.4$	GT-tuned	21.1	68.1	78.1	36.9	83.2	67.2
Fixed $\tau = 0.5$ (majority)	GT-tuned	16.5	79.8	63.5	26.3	89.7	50.7
Per-class threshold oracle	GT-tuned	39.4	—	—	—	—	—

LERF-OVS. On LERF-OVS, the same one-level rules are compared with IoU computed from the selection rendered to the evaluation frames as in Appendix F and averaged within each scene and then over scenes (Table 12). Without labels, the mixture segments close to the best fixed threshold ($\tau = 0.1$), returns empty predictions for slightly more no-target queries, and leaves no target query empty. Only $\tau = 0.15$ is better in both measures, while leaving 3.0% of the target queries empty; on ScanNet, the same threshold returns empty predictions for only 18.1% of the no-target queries, so no single fixed threshold is better in both measures on every benchmark.

Empty Predictions. Empty predictions come from the fitted boundary rather than from a better ranking (Table 13). The maximum score separates target from no-target queries about equally well with or without the mixture, but without it every observed Gaussian keeps a positive score and no prediction is empty. Removing the mixture from the full pipeline also lowers mIoU from 68.5 to 64.0 on LERF-OVS, from 42.5 to 42.3 on ScanNet, and from 95.3 to 74.3 on 3D-OVS. Feature aggregation becomes empty only when spatial averaging lowers its relevance below the fixed threshold, which also empties target queries (Table 14).

Table 13: Separating target from no-target queries with VerdictGS and without its mixture (otherwise the full pipeline): AUC of the maximum score over observed Gaussians, and N-acc (%).

	LERF-OVS		ScanNet		3D-OVS	
Configuration	AUC	N-acc	AUC	N-acc	AUC	N-acc
VerdictGS	0.879	35.1	0.831	54.3	0.951	79.8
Without mixture	0.877	0.0	0.857	0.0	0.958	0.0

Table 14: N-acc and T-acc for the feature- and response-based configurations (%). All rows except VerdictGS use medium-level inputs; feature rows select relevance above 0.5, the others scores above zero.

	LERF-OVS		ScanNet		3D-OVS	
Configuration	N-acc	T-acc	N-acc	T-acc	N-acc	T-acc
Features, no averaging	8.6	100.0	2.1	100.0	47.6	96.8
Features, spatial averaging	54.1	88.1	27.7	96.9	79.8	90.3
Response accumulation	0.0	100.0	0.0	100.0	0.0	100.0
+ Binomial mixture (medium)	41.6	100.0	59.6	86.5	82.3	90.3
VerdictGS	35.1	100.0	54.3	91.7	79.8	93.5

Target Queries Left Empty. On ScanNet, the single-level mixture leaves some target queries empty (Table 15), and its segmentation falls below response accumulation only in scenes where it leaves *floor* empty (Table 9). Multi-level inference recovers several of these queries, but *floor* remains empty in five of the ten scenes. In scene 0070, for example, the response rates for *floor* stay low across the scene (90th percentile 0.12–0.20 across levels), and the fitted target and background rates coincide at every level, so no target population forms.

Table 15: ScanNet target queries with an empty prediction. Feature aggregation leaves three target queries empty (0400 chair; 0590 sofa and refrigerator).

Scene	+ Binomial mixture (one level)	VerdictGS
0062	counter	counter
0070	floor	floor
0097	floor, counter, refrigerator, sink	floor, sink
0140	floor	floor
0200	floor, chair, door	floor, chair
0347	floor	floor
0400	chair	–
0590	refrigerator	–
Total	13	8

F 3D SELECTION ON LERF-OVS

Several works evaluate LERF-OVS by selecting Gaussians in 3D and rendering the selection, each with its own evaluation threshold (Appendix A). We apply such thresholds to the final field of VerdictGS and render the selection as in the evaluation of THGS: all Gaussians are rendered with the selected ones set to one, and pixels above 0.5 form the prediction.

With VALA's per-Gaussian rule, VerdictGS reaches 64.1 at VALA's threshold and 64.1–66.0 across the relative thresholds of Table 16, numerically above the 3D results reported for VALA, Occam's LGS, and VoteSplat, although their renderings and scenes differ. These values lie 2.5–4.4 below the official 2D evaluation (68.5), which sets its cut in each frame and smooths the map, which a single 3D selection cannot.

Table 16: 3D-selection mIoU (%) on LERF-OVS. Reported results are from Table 1 of VALA (Wang et al., 2026), which evaluated VALA and Occam's LGS (Cheng et al., 2025) and took VoteSplat (Jiang et al., 2025) from its paper; each follows its own rendering and binarization. VerdictGS applies VALA's per-Gaussian selection rule (KNN smoothing, min–max normalization, threshold) or a cut at half of the maximum score to its final field and renders the selection as in the evaluation of THGS, so the rows are not directly comparable (Appendix A).

Method	figurines	ramen	teatime	waldo	Mean
Occam's LGS	52.9	32.0	61.0	43.0	47.2
VoteSplat	68.6	39.2	66.7	25.8	50.1
VALA	60.4	45.4	70.6	55.7	58.0
VerdictGS, VALA's rule, threshold 0.6	67.6	60.5	74.6	53.9	64.1
VerdictGS, VALA's rule, threshold 0.4	70.3	59.9	74.0	57.2	65.3
VerdictGS, half of the maximum	70.0	60.8	74.0	58.9	66.0

G MULTI-LEVEL INFERENCE

This section supports the analysis of multi-level inference in Sec. 4.4 and the design of Sec. 3.4.

Single Levels. No single mask level is best on every benchmark, and fusing the four levels exceeds each of them on all three (Table 17). Medium is the strongest level on LERF-OVS and large on ScanNet, whose classes cover whole objects and surfaces, so a fixed choice of level would favor one benchmark over the other.

Table 17: Single levels and level fusion (mIoU, %). All rows except VerdictGS fit the mixture without propagation and use ordinary final averaging; VerdictGS adds gated response propagation and membership-conditioned averaging.

Levels	LERF-OVS	ScanNet	3D-OVS
Default only	61.2	38.1	89.9
Small only	57.1	22.3	82.6
Medium only	64.6	34.6	89.7
Large only	58.3	39.5	90.3
Four levels, fused	66.5	40.8	**95.3**
VerdictGS	**68.5**	**42.5**	**95.3**

Propagation and Final Averaging. Gated response propagation and membership-conditioned averaging both recover missed parts of the target, and the two gates keep this recovery from spreading onto other objects (Table 18). Propagating a region's response to its parts raises recall, whereas removing either gate leaves recall similar or higher but lowers precision, and without the query gate 3D-OVS drops the most. Relative to no final averaging, ordinary mask averaging trades recall for precision, whereas averaging separately over target and background Gaussians raises both.

Table 18: Propagation and final averaging. mIoU, recall (Rec.), and precision (Prec.) in %, computed as in Table 8; each row changes the stated steps of VerdictGS, shown in the last row.

		LERF-OVS			ScanNet	3D-OVS
Propagation	Final averaging	mIoU	Rec.	Prec.	mIoU	mIoU
none	ordinary	66.5	77.0	77.6	40.8	**95.3**
none	membership-conditioned	68.0	80.0	76.8	40.9	**95.3**
gated	none	66.6	79.6	75.4	40.7	94.5
gated	ordinary	67.0	78.4	76.7	42.2	**95.3**
without semantic gate	membership-conditioned	66.7	81.4	74.0	42.4	95.2
without query gate	membership-conditioned	66.3	82.8	73.2	40.7	90.1
gated	membership-conditioned	**68.5**	81.4	75.9	**42.5**	**95.3**

Propagation Order. Which levels are linked matters little, but how the responses pass between them does (Table 19). All five pairings that propagate stay within a narrow range above no propagation, whereas a one-pass rule that lets every mask inherit from a larger mask in any other level loses the gain on ScanNet.

Table 19: Propagation pairings (mIoU, %). Arrows give the source and receiving levels; the default setting is the first row.

Propagation	LERF-OVS	ScanNet	3D-OVS	Mean
large, default → medium; medium → small	68.5	42.5	95.3	68.7
large → medium → small	68.3	41.8	95.3	68.5
large, default → medium	67.9	42.4	95.3	68.5
large → medium	67.9	41.7	95.3	68.3
default → medium → small	68.9	42.3	95.3	68.8
Every level pair, one pass, larger source only	67.8	41.1	95.3	68.1
No propagation	68.0	40.9	95.3	68.1

H Efficiency Details

Measurement. Table 3 is measured on figurines on one machine (NVIDIA RTX 6000 Ada, AMD EPYC 7313). Baselines run their released code from the same cached SAM and CLIP features. Preparation is the wall-clock time from inputs in memory, including loading and saving, and excludes reconstruction and feature extraction. Query time is the warm time from CLIP text encoding to per-Gaussian results, averaged over 21 queries; for Dr. Splat, which reads pre-encoded text, we add the measured encoding time.

VerdictGS. The CPU builds the caches in 38 s from contribution records that a 2DGS rasterizer, modified to accumulate contributions per mask, renders for 299 views in 8 s on the GPU; these GPU records give the same figurines mIoU within 0.01. Table 20 breaks down the query time; GPU and CPU give the same LERF-OVS mIoU.

Table 20: Per-query time (ms) of VerdictGS on figurines.

	Text encoding	Inference	Total
GPU	5	11	16
CPU, 1 thread	20	472	492
CPU, 8 threads	10	204	214

I Additional Qualitative Results

LERF-OVS. Fig. 7 extends Fig. 3(a) to all four scenes. Where THGS selects superpoints that miss the target in the frame, VerdictGS covers the target, from whole objects such as the *bowl* and the *cabinet* to small ones such as *waldo*.

Targets of Different Sizes. Fig. 8 shows queries of different sizes in one ramen frame: the bowl and four items inside it. VerdictGS selects each target at its own extent, from the whole bowl to a single slice of *kamaboko*, whereas THGS can return only its predefined superpoints, which here miss the bowl and merge the *kamaboko* with the surrounding contents of the bowl.

ScanNet. Fig. 9 shows the class assigned to each vertex. Since the THGS code evaluates only LERF-OVS, we replay THGS from its public superpoint hierarchy: each vertex takes the nearest THGS Gaussian, and each class scores the larger relevance of that Gaussian's level-2 and level-3 superpoints, computed as in its LERF-OVS evaluation. This replay reaches 34.7 mIoU, close to the reported 34.4, and VerdictGS is higher in seven of the ten scenes. THGS labels whole superpoints, so a wall can take the class of a picture, door, or sink on it, whereas VerdictGS scores each Gaussian and keeps walls and floors apart from the objects on them.

Selection of Fig. 3. For its target queries, each example is the frame–query pair whose IoU gain over THGS is closest to the mean gain among the pairs where VerdictGS is better. For its no-target queries, the examples are the two queries for which VerdictGS selects no observed Gaussian and the THGS selection is largest in the frame.

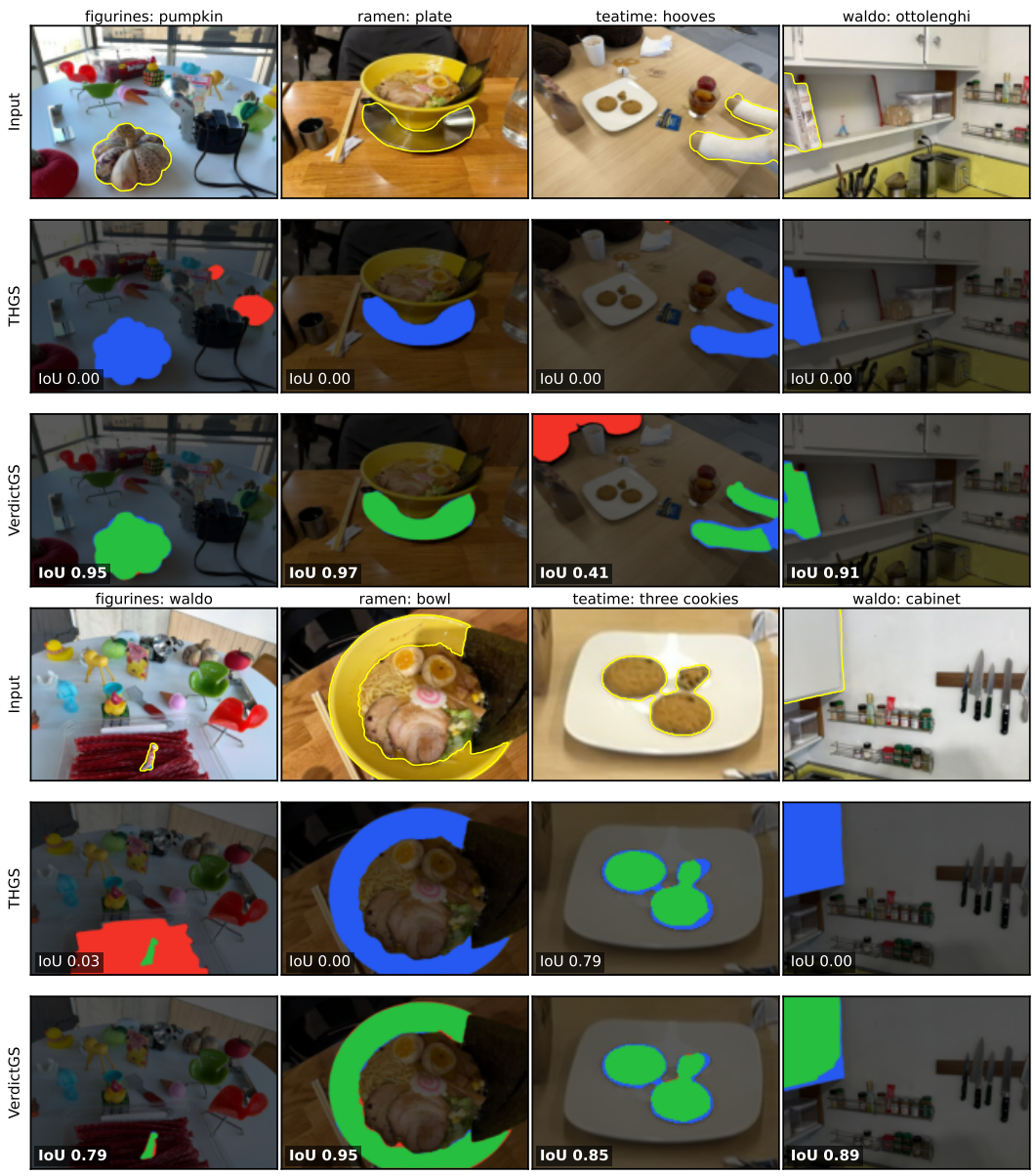

Figure 7: Comparison with THGS on LERF-OVS under the LangSplat evaluation, as in Fig. 3(a) (yellow: ground truth; green: correct; blue: missed; red: outside the target). For each scene, the two frame–query pairs with the largest IoU gain over THGS are shown, at most one per query.

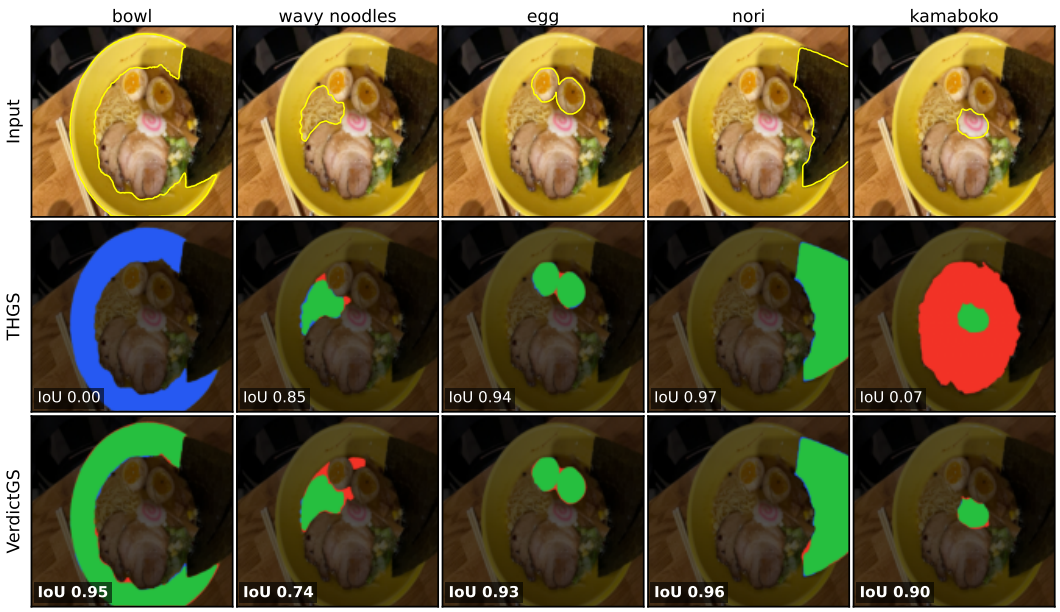

Figure 8: Queries of different sizes in one ramen frame, the bowl and four items inside it, under the LangSplat evaluation (colors as in Fig. 7).

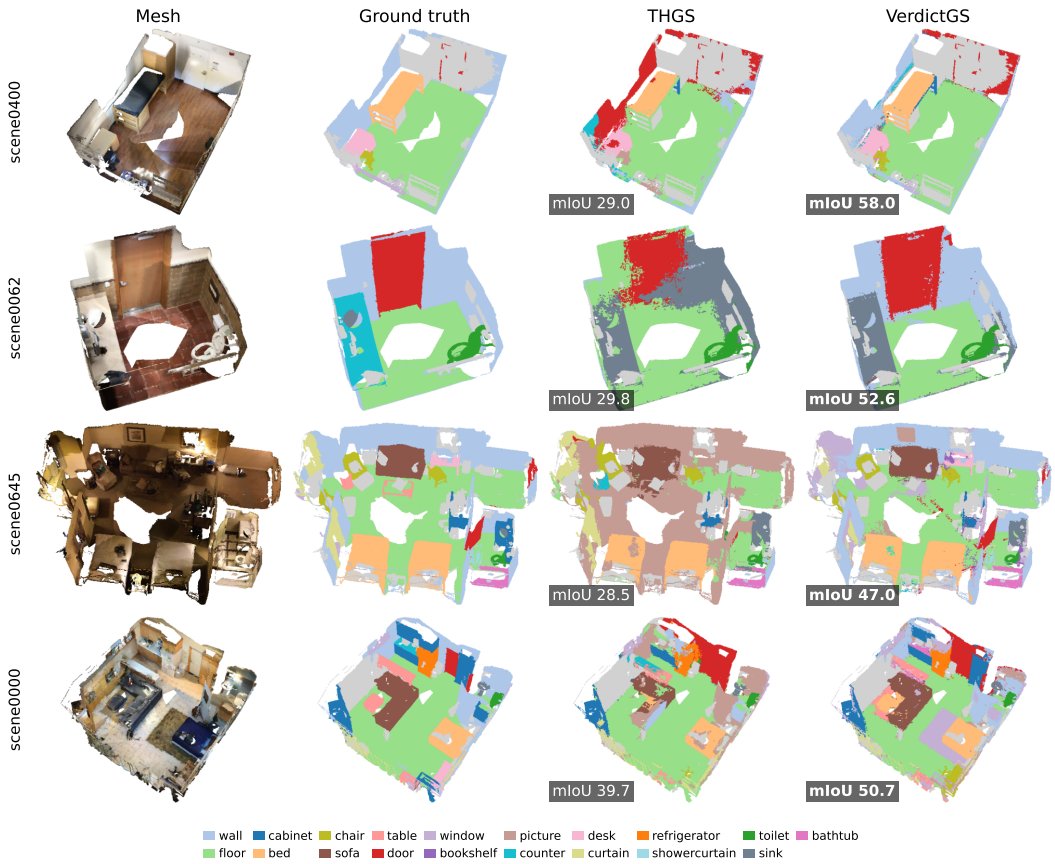

Figure 9: Comparison with THGS on ScanNet (19 classes): each vertex is colored by its highest-scoring class, and vertices outside the 19 classes are gray. The four scenes with the largest mIoU gain over THGS are rendered from the same oblique view, with back faces removed to show the interior.