B^3-Seg: Camera-Free, Training-Free 3DGS Segmentation via Analytic EIG and Beta–Bernoulli Bayesian Updates

Hiromichi Kamata
Sony Group Corporation
Tokyo, Japan
Hiromichi.Kamata@sony.com

Samuel Arthur Munro
Pixomondo
London, United Kingdom
samuelm@pixomondo.com

Fuminori Homma
Sony Group Corporation
Tokyo, Japan
Fuminori.Homma@sony.com

Abstract

*Interactive 3D Gaussian Splatting (3DGS) segmentation is essential for real-time editing of pre-reconstructed assets in film and game production. However, existing methods rely on predefined camera viewpoints, ground-truth labels, or costly retraining, making them impractical for low-latency use. We propose **B^3-Seg (Beta–Bernoulli Bayesian Segmentation for 3DGS)**, a fast and theoretically grounded method for open-vocabulary 3DGS segmentation under **camera-free** and **training-free** conditions. Our approach reformulates segmentation as sequential Beta–Bernoulli Bayesian updates and actively selects the next view via analytic Expected Information Gain (EIG). This Bayesian formulation guarantees the adaptive monotonicity and submodularity of EIG, which produces a greedy $(1-1/e)$ approximation to the optimal view sampling policy. Experiments on multiple datasets show that B^3-Seg achieves competitive results to high-cost supervised methods while operating end-to-end segmentation within a few seconds. The results demonstrate that B^3-Seg enables practical, interactive 3DGS segmentation with provable information efficiency.*

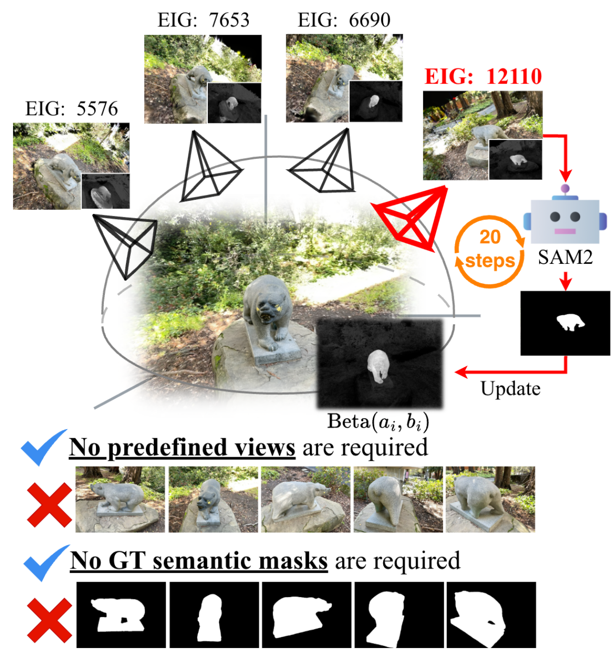

Figure 1. B^3-Seg actively selects the next view via Expected Information Gain and updates 3D labels by Beta–Bernoulli Bayesian updates. It runs in a few seconds and requires neither predefined views nor ground-truth semantic labels.

1. Introduction

Recently, 3D Gaussian Splatting (3DGS) [12] has rapidly gained attention as a 3D representation that combines real-time rendering with high visual fidelity. In film and game production, it is increasingly common that only a pre-reconstructed 3DGS asset is shared across teams. Thus, interactive 3DGS segmentation—selecting, editing, and removal of objects directly on the asset—has become a required capability [16, 26].

Many recent works tackle open-vocabulary 3DGS segmentation and achieve strong accuracy [18, 19, 28, 29, 34]. However, most methods assume access to predefined camera viewpoints and/or ground-truth semantic masks, which does not align with the practical setting where only a shared 3DGS asset is available. In real use, interactive editing must operate camera-free, training-free, and open-vocabulary, and return results within a few seconds.

Interactive editing also requires low latency. Although several recent methods [18, 28, 29, 34] are highly accurate, they are based on large-scale pretraining and often take minutes to tens of minutes per scene. Faster approaches that return results within a minute have also appeared [23, 30], but still assume access to camera viewpoints and ground-truth labels.

We introduce **B^3-Seg (Beta–Bernoulli Bayesian Seg-**

mentation for 3DGS). B^3-Seg runs under **camera-free, training-free, open-vocabulary** conditions and produces results within **a few seconds**. We reformulate 3DGS segmentation as sequential Beta–Bernoulli Bayesian updates and select the next best view via analytic Expected Information Gain (EIG). On each actively selected view, we obtain masks using Grounding DINO + SAM2 [22] with CLIP [20] re-ranking, and we update the Beta posteriors accordingly. Building on this Bayesian formulation, we prove the non-negativity and diminishing returns of EIG, and derive a greedy $(1-1/e)$ approximation guarantee. This yields an interactive 3DGS segmentation method that is both information-efficient and theoretically grounded.

Our contributions are as follows.

- **Camera-free, training-free, and few-second segmentation.** We achieve open-vocabulary 3DGS segmentation in a few seconds without camera viewpoints, ground-truth labels, or retraining.
- **Bayesian reformulation.** We reformulate 3DGS segmentation as sequential Beta–Bernoulli Bayesian updates, providing a unified and robust probabilistic model.
- **Analytic EIG and active view selection.** We estimate per-view pseudo-counts from one render and choose the next view using analytic EIG.
- **Theoretical guarantees.** We show non-negativity and diminishing returns of EIG, confirming adaptive monotonicity and adaptive submodularity, leading to greedy $(1-1/e)$ approximation.
- **Competitive accuracy.** Across datasets, our method is competitive with slower, label-dependent baselines while preserving the practical constraints above.

2. Related Works

2.1. 3DGS Segmentation

Recent works improve 3DGS with semantics through text conditioning and cross-modal supervision, allowing open-vocabulary retrieval and segmentation [13, 19, 28, 29, 34]. LERF [13] integrates CLIP features into a NeRF by optimizing a dense language field across views, enabling flexible text queries but requiring access to multi-view images and extended scene optimization. Gaussian Grouping [29] effectively groups Gaussians into instances/parts using multi-view supervision and feature aggregation, achieving high accuracy with additional optimization for all reconstruction views. OpenGaussian [28] distills open-vocabulary 2D segmenters into 3D Gaussians using cross-modal pretraining and multi-stage pipelines with multi-view images and camera trajectories. ObjectGS [34] creates an object-centric 3DGS with instance-level reasoning and compositional editing. However, it requires strong semantic priors, and extensive computation for per-scene optimization. These methods achieve high accuracy, but their need for reconstruction images/camera paths, ground-truth or distilled labels, and significant optimization time limits their use in interactive editing with only a stand-alone 3DGS asset.

2.2. Few-Second 3DGS Segmentation

To improve responsiveness, few-second frameworks such as FlashSplat [23] and COB-GS have been proposed [30]. FlashSplat casts the consistency between 2D masks and 3D labels as a linear program, and COB-GS maintains quality via boundary-aware improvements and texture heuristics. However, these approaches still assume the availability of reconstruction views and semantic masks, and they provide no theoretical guarantees. iSegMan [31] presents an interactive segmentation framework for 3DGS scenes. It uses visibility voting from multiple views based on empirically designed camera trajectories.

2.3. Active View Sampling on 3DGS

Active view selection has been widely studied for 3D reconstruction. FisherRF [10] formulates the next-best view using Fisher information. ActiveGS [11] uses 3DGS as a map representation to greedily select the next-best view based on confidence and coverage. ActiveGAMER [5] leverages fast 3DGS rendering and an information-gain objective with coarse-to-fine search to improve fidelity and completeness. ActiveSGM [6] integrates active camera sampling and segmentation in indoor scenes such as Replica [25], but requires backpropagation-based training and targets a fixed set of classes via OneFormer [8]. In contrast, our method provides Bayesian updates with analytic EIG, and operates camera-free and training-free under open-vocabulary conditions.

3. Proposed Method

B^3-Seg performs a few-second and accurate segmentation of a user-specified object from a 3DGS scene under camera-free and training-free conditions. The method has two components: (i) 3DGS segmentation via sequential Beta–Bernoulli Bayesian updates of per-Gaussian probabilities, and (ii) active view selection based on the EIG in Beta distributions.

3.1. Preliminaries

3DGS [12] represents a scene as a set of Gaussians $\mathcal{G}=\{g_i\}_{i=1}^{N}$. Each Gaussian g_i has mean μ_i, covariance Σ_i, opacity α_i, and color c_i. Rendering from a camera view v accumulates per-Gaussian contributions as

$$I(v) = \sum_i c_i\, \alpha_i\, T_i, \tag{1}$$

where $T_i = \prod_{j=1}^{i-1}(1-\alpha_j)$ denotes the transmittance along the ray in view v.

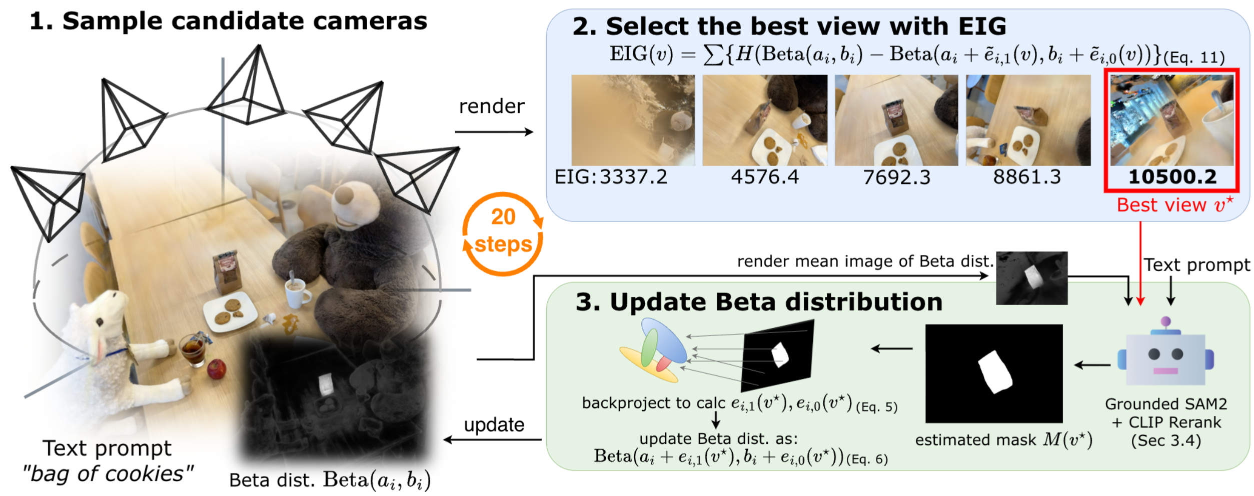

Figure 2. **Overview of B^3-Seg**. (1) Sample N_{cand} candidate views on a sphere centered at the estimated object center $\mathbf{c}_{\mathrm{obj}}$. (2) Render each candidate to compute $\mathrm{EIG}(v)$ by Eq. (11), and pick the best view $v^\star$ (red). On $v^\star$, obtain masks using Grounded SAM2 and CLIP reranking. (3) From the mask, compute $(e_{i,1}, e_{i,0})$ by Eq. (5) and update Beta parameters. We iterate (1)–(3) process 20 steps. The pipeline enables camera-free, training-free, open-vocabulary 3DGS segmentation in a few seconds.

FlashSplat [23] uses linear programming to solve binary assignments $P_i \in \{0, 1\}$, determining whether g_i is part of the target object or not.

$$\min_{\{P_i\}} \sum_v \sum_{(j,k)\in I(v)} \left| \sum_i P_i\, \alpha_i\, T_i \;-\; M_{j,k}(v) \right|, \tag{2}$$

where $M(v)$ is the binary mask of $I(v)$. FlashSplat produces the following decision rule:

$$\begin{aligned} P_i &= \arg\max_n A_{i,n}, \quad n \in \{0,1\}, \\ \text{where} \quad A_{i,n} &= \sum_v \sum_{(j,k)\in I(v)} \alpha_i T_i\, \mathbb{I}[M_{j,k}(v) = n], \end{aligned} \tag{3}$$

which compares visible responsibility inside and outside the mask. COB-GS [30] uses a similar gradient descent update.

3.2. Bayesian Reformulation of 3DGS Segmentation

We use the insights of Sec 3.1 to reformulate 3DGS segmentation as probabilistic Bayesian updates. Let $y_i \in \{0, 1\}$ indicate whether Gaussian g_i belongs to the user-specified object. We place a Bernoulli–Beta prior/posterior on $p_i = \Pr(y_i{=}1)$:

$$y_i \mid p_i \sim \mathrm{Bernoulli}(p_i), \qquad p_i \sim \mathrm{Beta}(a_i, b_i). \tag{4}$$

Given a rendered image $I(v)$ and an object mask $M(v)$, we treat per-pixel responsibilities as observations of success counts $e_{i,1}(v)$ and failure counts $e_{i,0}(v)$:

$$\begin{aligned} e_{i,1}(v) &= \sum_{(j,k)\in I(v)} \alpha_i T_i\, \mathbb{I}[M_{j,k}(v) = 1], \\ e_{i,0}(v) &= \sum_{(j,k)\in I(v)} \alpha_i T_i\, \mathbb{I}[M_{j,k}(v) = 0]. \end{aligned} \tag{5}$$

By Beta-Bernoulli conjugacy, the posterior updates are

$$\mathrm{Beta}(a_i, b_i) \leftarrow \mathrm{Beta}(a_i + e_{i,1}(v), b_i + e_{i,0}(v)). \tag{6}$$

After multiple views, the posterior is

$$p_i \sim \mathrm{Beta}\Big(a_{\mathrm{init}} + \textstyle\sum_v e_{i,1}(v),\ b_{\mathrm{init}} + \sum_v e_{i,0}(v)\Big). \tag{7}$$

For $a_{\mathrm{init}}{=}b_{\mathrm{init}}$, the posterior mean becomes $a_i/(a_i{+}b_i)$. In this symmetric case, the Bayes-optimal label reduces to selecting the class with the larger accumulated pseudo-counts, which exactly recovers the decision rule of Eq. (3):

$$\begin{aligned} y_i &= \arg\max_{n\in\{0,1\}} \sum_v e_{i,n}(v) \\ &= \arg\max_{n\in\{0,1\}} \sum_v \sum_{(j,k)\in I(v)} \alpha_i T_i\, \mathbb{I}[M_{j,k}(v) = n]. \end{aligned} \tag{8}$$

Thus, FlashSplat's label-selection rule appears as the MAP decision within our Bayesian formulation.

3.3. Active Camera Sampling Based on EIG

To efficiently estimate p_i, we sample the most informative camera view from randomly sampled candidate views. The details of candidate view sampling are described in Sec 3.5. For each candidate view v, the information gain is the difference in entropy before and after adding that view:

$$\begin{aligned} \mathrm{IG}(v) = \sum_i \{ & H(\mathrm{Beta}(a_i, b_i)) \\ & - H\left(\mathrm{Beta}(a_i + e_{i,1}(v), b_i + e_{i,0}(v))\right)\} \end{aligned} \tag{9}$$

where H denotes entropy and the Beta entropy can be calculated analytically (Appendix A). However, due to the need

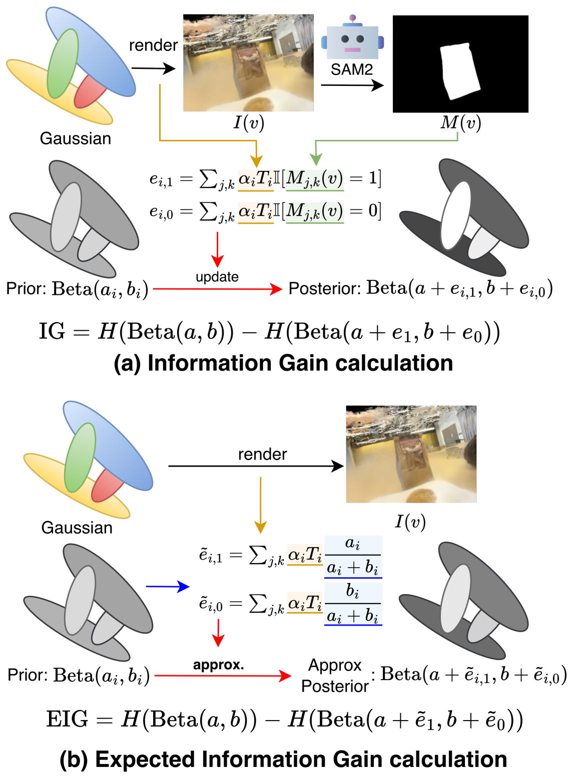

Figure 3. **Information Gain vs. Expected Information Gain (ours).** (a) IG calculation updates the Beta posterior using SAM2 segmentation masks (Eq. (9)). (b) Our EIG approximates the posterior update from the prior Beta distribution, avoiding SAM2 inference and enabling efficient viewpoint evaluation (Eq. (11)).

for mask estimation, it is inefficient to calculate $\mathrm{IG}(v)$ for all candidate views. Instead, we use **Expected Information Gain (EIG)**. The per-Gaussian responsibility in the rendered image is calculated as $\tau_i = \sum_{(j,k)\in I(v)} \alpha_i T_i$, and success/failure counts can be approximated using the mean of the Beta distribution $m_i = a_i/(a_i + b_i)$.

$$\tilde{e}_{i,1} = m_i \tau_i = \sum_{(j,k)\in I(v)} \alpha_i T_i \frac{a_i}{a_i + b_i},$$
$$\tilde{e}_{i,0} = (1 - m_i)\tau_i = \sum_{(j,k)\in I(v)} \alpha_i T_i \frac{b_i}{a_i + b_i}. \tag{10}$$

Assuming g_i falls inside the mask with probability m_i, Eq. (10) approximates Eq. (9). From them, EIG is calculated as follows:

$$\mathrm{EIG}(v) = \sum_i \{H(\mathrm{Beta}(a_i, b_i)) - H\left(\mathrm{Beta}(a_i + \tilde{e}_{i,1}(v), b_i + \tilde{e}_{i,0}(v))\right)\} \tag{11}$$

Then, we greedily select the view with the highest EIG,

$$v^\star = \arg\max_v \mathrm{EIG}(v) \tag{12}$$

Algorithm 1 B^3-Seg: Camera-Free, Training-Free 3DGS Segmentation via Analytic EIG

Initialize $(a_i, b_i) \leftarrow (a_{\text{init}}, b_{\text{init}})$ for all Gaussians.
From a canonical view of the scene, estimate an initial mask using segmentation module of Sec. 3.4.
Update (a_i, b_i) with Eq. (6).
Compute $\mathbf{c}_{\text{obj}}, r_{\text{obj}}$ from Gaussians satisfying $a_i > b_i$.
for iteration $t = 1, \ldots, T$ **do**
  Uniformly sample N_{cand} candidate viewpoints on the sphere centered at $\mathbf{c}_{\text{obj}}$, with a radius r_{sphere}.
  for each candidate v **do**
    Render once to obtain $\tau_i = \sum_{(j,k)} \alpha_i T_i$.
    Compute $(\tilde{e}_{i,1}, \tilde{e}_{i,0})$ with Eq. (10).
    Compute $\mathrm{EIG}(v)$ with Eq. (11).
  end for
  Select $v^\star = \arg\max_v \mathrm{EIG}(v)$.
  Render RGB image $I(v^\star)$.
  Estimate $M(v^\star)$ with the module of Sec. 3.4.
  Compute $(e_{i,1}(v^\star), e_{i,0}(v^\star))$ with Eq. (5).
  Beta update: $(a_i, b_i) \leftarrow (a_i + e_{i,1},\ b_i + e_{i,0})$.
  Update $\mathbf{c}_{\text{obj}}, r_{\text{obj}}$ from Gaussians with $a_i > b_i$.
end for
Return final 3D mask: Gaussians with $a_i > b_i$

The stability and validity of EIG-based greedy sampling are discussed in Section 4.

3.4. Open-Vocabulary Mask Inference

To obtain a 2D semantic mask for each selected view, we adopt a lightweight open-vocabulary segmentation module. This module consists of three stages:

1. **Text-conditioned region proposal (Grounding DINO).** Given a user prompt, Grounding DINO [15] generates candidate bounding boxes $\{B_k\}$ that indicate possible object regions.
2. **Mask prediction with prior guidance (SAM2).** For each candidate box B_k in a view $v^\star$, we use SAM2 [21] for segmentation masks. To stabilize inference, SAM2's `mask_input` receives a prior image $R(v^\star)$, rendered from current Beta means $m_i = \frac{a_i}{a_i+b_i}$:

$$R_{\text{soft}}(v^\star) = \sum_i m_i\, \alpha_i\, T_i(v^\star),\ R(v^\star) = \log \frac{R_{\text{soft}}(v^\star)}{1 - R_{\text{soft}}(v^\star)}.$$

   This cues SAM2 with view-specific information from Beta posterior means, ensuring temporal consistency and reducing drift to distractors.

$$M_k(v^\star) = \mathrm{SAM2}\left(I(v^\star),\ B_k,\ \texttt{mask\_input} = R(v^\star)\right).$$

3. **Semantic ranking via CLIP.** Each candidate mask $M_k(v)$ is applied to the RGB image to obtain a masked

crop, which is scored by CLIP [20] against the user text. The highest scoring mask for the view v is chosen:

$$M(v^\star) = \arg\max_k \text{CLIP}\big(I(v^\star) \odot M_k(v^\star),\ \text{text}\big).$$

Following the selected mask, $e_{i,1}(v^\star)$ and $e_{i,0}(v^\star)$ are calculated with Eq. (5). We then update the Beta parameters according to Eq. (6):

$$\text{Beta}(a_i, b_i) \leftarrow \text{Beta}(a_i + e_{i,1}(v^\star), b_i + e_{i,0}(v^\star)).$$

We repeat this view-selection and update loop.

3.5. Overall Pipeline and Candidate Sampling

Algorithm 1 summarizes the pipeline. We initialize all Gaussians with $(a_{\text{init}}, b_{\text{init}})$ and obtain an initial mask from a canonical view using the procedure in Section 3.4. After this first update, we consider Gaussians with $a_i > b_i$ as foreground and estimate the object center and radius by

$$\mathbf{c}_{\text{obj}} = \frac{\sum_{i\in\mathcal{G}_{\text{fg}}} m_i\,\mu_i}{\sum_{i\in\mathcal{G}_{\text{fg}}} m_i},\ r_{\text{obj}} = \frac{\sum_{i\in\mathcal{G}_{\text{fg}}} m_i\,\|\mu_i - \mathbf{c}_{\text{obj}}\|}{\sum_{i\in\mathcal{G}_{\text{fg}}} m_i}.$$

Then, N_{cand} candidate cameras are uniformly sampled from the $\mathbf{c}_{\text{obj}}$ centered sphere, with a radius calculated as $r_{\text{sphere}} = 1.5\, r_{\text{obj}}/\tan(\text{fov}/2)$. Next, we compute the EIG for each candidate and pick the most informative view. We repeat mask inference and updates (a_i, b_i), while updating $\mathbf{c}_{\text{obj}}, r_{\text{obj}}$ at each iteration. The final 3D mask consists of Gaussians with $a_i > b_i$.

4. Theoretical Guarantees

This section shows that the proposed method is (i) adaptive monotone and (ii) adaptive submodular, and then derives (iii) a greedy $(1{-}1/e)$ approximation guarantee. Let S be the set of already selected views, and denote by $\text{EIG}(v \mid S)$ the conditional expected gain when adding a candidate view v under S (the per-view EIG itself is given in Eq. (11)). The detailed proofs are provided in the Appendix.

Lemma 1 (Adaptive monotonicity). $\text{EIG}(v \mid S)$ is adaptive monotone, i.e., $\mathbb{E}[\text{EIG}(v \mid S)] \geq 0$ for any selected set S and any candidate v.
Sketch. The Beta entropy $H(\text{Beta}(a_i, b_i))$ is non-increasing in $\kappa_i = a_i{+}b_i$ for $\kappa_i \geq 2$ (Appendix. B), which is satisfied by our initialization ($a_{\text{init}} = b_{\text{init}} = 1$). Each candidate v adds nonnegative pseudo-counts $\tau_i(v) \geq 0$ to κ_i, so the posterior entropy decreases in expectation, which yields $\mathbb{E}[\text{EIG}(v \mid S)] \geq 0$. Intuitively, observing an additional view always decreases uncertainty, so the learning progress is stable.

Lemma 2 (Adaptive submodularity). $\text{EIG}(v \mid S)$ is adaptive submodular, i.e., $\mathbb{E}[\text{EIG}(v \mid S)] \geq \mathbb{E}[\text{EIG}(v \mid S')]$ for any $S' \supseteq S$ and any candidate v.
Sketch. As we observe more views, each κ_i increases and the reduction in Beta entropy (Eq. (11)) becomes smaller (Appendix. C). In addition, $\tau_i(v)$ is added linearly and is nonnegative. Therefore, the expected marginal gain for the same v is smaller under the larger set S'.

Theorem (Greedy $(1{-}1/e)$ approximation). At each step t, we greedily select $v_t^{\text{greedy}} = \arg\max_v \text{EIG}(v \mid S_{t-1})$, where S_{t-1} is the set of views already selected. We define the total expected information gain of this greedy policy as

$$\text{EIG}(S_k^{\text{greedy}}) := \sum_{t=1}^{k} \text{EIG}(v_t^{\text{greedy}} \mid S_{t-1}^{\text{greedy}}).$$

Then, according to Theorem 16 of [7],

$$\mathbb{E}\Big[\text{EIG}(S_k^{\text{greedy}})\Big] \;\geq\; (1 - 1/e)\ \max_{\pi}\ \mathbb{E}[\text{EIG}(S_k^{\pi})]\,, \quad (13)$$

where S_k^{π} denotes the sets obtained by all adaptive policies. Thus, our EIG-based greedy selection achieves a $(1{-}1/e)$ approximation to the optimal view selection policy.

5. Experiment

5.1. Experimental Settings

We evaluated on two datasets: LERF-Mask [29] and 3D-OVS [14]. For LERF-Mask, we follow the Gaussian Grouping protocol [29]; for 3D-OVS, we follow [34]. We set the number of update iterations to $T{=}20$, the number of candidate views to $N_{\text{cand}}{=}20$, and initialize the Beta parameters with $a_{\text{init}}{=}b_{\text{init}}{=}1$. Each sequence starts with a single canonical scene view, providing the initial mask and object center. The canonical view is the first camera viewpoint in the scene metadata, which does not need extra supervision or specific reconstruction data—just the camera pose of dataset. It is used only to obtain the initial prior, with subsequent views chosen by our EIG-based planning. The low dependency on the initial condition is discussed in Sec. 5.4 Our experiments were conducted with a single RTX A6000 GPU. The end-to-end runtime (rendering, mask inference, and updates) is within a few seconds, enabled by our approximation and active selection (Section 4).

5.2. Qualitative Results

Figure 4 shows representative comparisons in the LERF-Mask and 3D-OVS dataset. Our B^3-Seg produces cleaner, more complete object masks than prior 3DGS methods; for example, it recovers the full chair in the "green toy chair" scene and segments the "stuffed bear" despite heavy clutter. These improvements follow from two complementary components: analytic EIG that prioritizes highly informative views, and robust Beta-Bernoulli updates that accumulate reliable pseudo-counts from the 2D mask. Importantly,

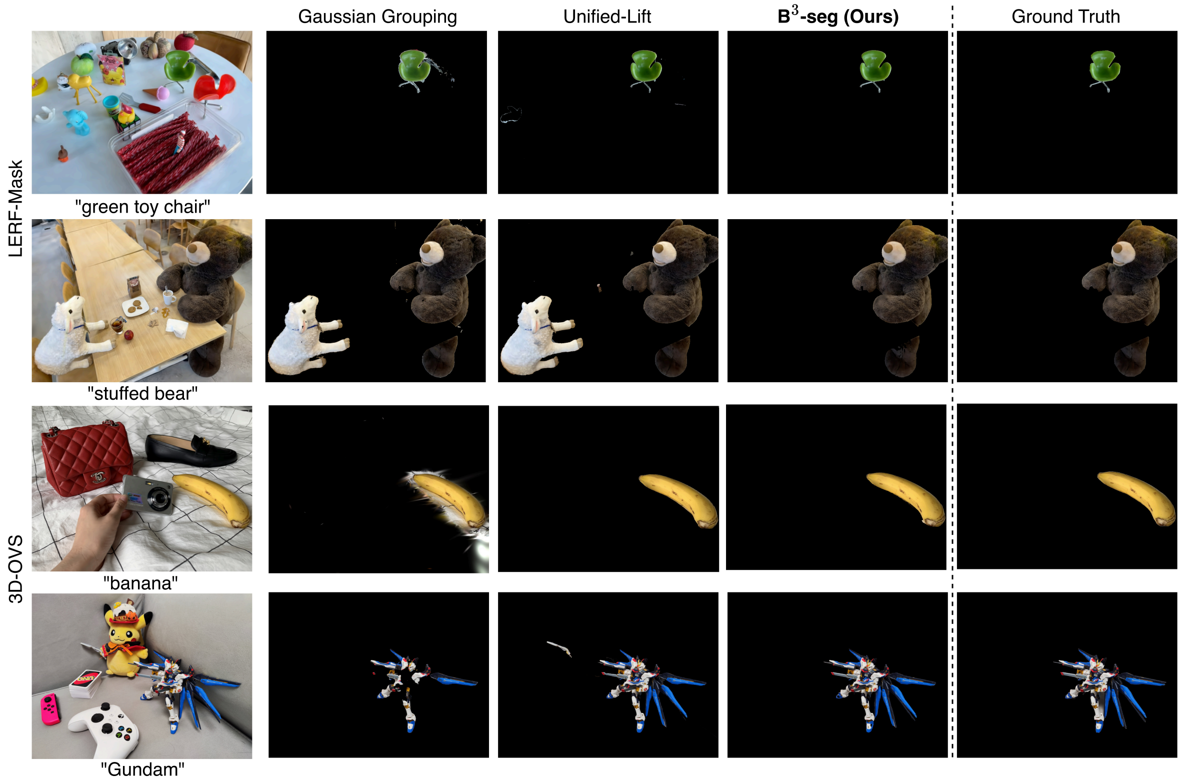

Figure 4. Qualitative comparison on text-guided 3D segmentation. We compare our method (**B^3-Seg**) with prior 3DGS segmentation approaches. Our method produces cleaner and more complete object masks, especially in cluttered scenes.

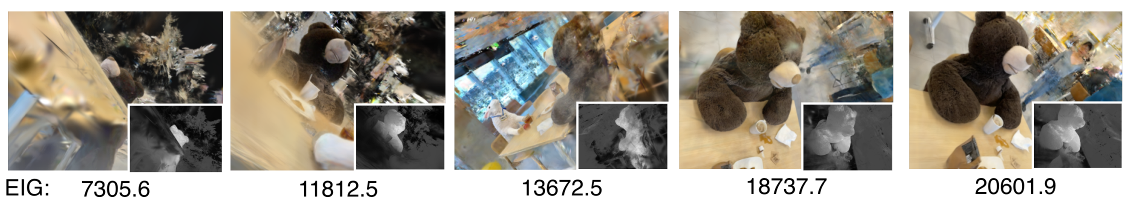

Figure 5. Candidate-view EIG on LERF-Mask (Teatime) with the prompt *"stuffed bear"*. Each panel shows a candidate rendering; the bottom-right inset is the current confidence map (posterior mean).

almost high-mIoU baselines rely on reconstruction views, ground-truth masks, or per-scene optimization that take tens of minutes. By contrast, our B^3-Seg achieves comparable visual quality in a budget of 20 views and a few seconds.

Figure 5 illustrates our view selection on the LERF-Mask Teatime scene. Views where the bear is larger and less occluded have a higher EIG, showing that our view-selection criterion selects views that reduce uncertainty.

5.3. Quantitative Results

LERF-Mask. Table 1 reports accuracy and latency. As baselines, FlashSplat (Uniform-Sphere) and FlashSplat (Recon-Cam) are used to isolate sampling effects. In FlashSplat (Uniform-Sphere), N_{cand} candidate viewpoints are sampled uniformly on a sphere centered at the estimated object center; a random candidate chosen (no camera priors, no EIG). In FlashSplat (Recon-Cam), N_{cand} candidate viewpoints are randomly sampled from reconstruction cameras (available on LERF-Mask). Both baselines used the same segmentation pipeline with B^3-seg for a fair comparison. From Table 1, B^3-Seg achieves higher segmentation scores than baselines and is competitive even with methods that assume reconstruction views.

3D-OVS. Table 2 follows the same presentation: the top block lists methods that assume reconstruction views/labels, while the bottom block compares sampling-based, training-free approaches under the 20-view budget. As shown in Table 2, our B^3-Seg surpasses baselines in segmentation scores and matches the performance of methods with reconstruction views.

Method	Accuracy (mIoU / mBIoU)				views	time	steps
	figurines	ramen	teatime	mean			
Assumes reconstruction views and/or labels (not directly comparable)							
LERF [13]	33.5 / 30.6	28.3 / 14.7	49.7 / 42.6	37.2 / 29.3	GT	45 min	30k
SA3D [2]	24.9 / 23.8	7.4 / 7.0	42.5 / 39.2	24.9 / 23.3	GT	35 min	30k
LangSplat [19]	52.8 / 50.5	50.4 / 44.7	69.5 / 65.6	57.6 / 53.6	GT	19 min	30k
Gaussian Grouping [29]	69.7 / 67.9	77.0 / 68.8	71.7 / 66.1	72.8 / 67.6	GT	37 min	30k
Gaga [17]	**90.7 / 89.0**	64.1 / 61.6	69.3 / 66.0	74.7 / 72.2	GT	13 min	30k
Unified-Lift [33]	–	–	–	80.9 / 77.1	GT	40 min	30k
ObjectGS [34]	88.2 / 89.0	**88.0 / 79.9**	**88.9 / 88.6**	**88.4 / 85.8**	GT	$\sim$ 50 min	30k
Sampling-based, no retraining (directly comparable within this block)							
FlashSplat [23] (Uniform-Sphere)†	60.2 / 57.5	68.4 / 61.5	80.4 / 76.3	69.6 / 65.1	Sample	**10.2 sec**	**20**
FlashSplat [23] (Recon-Cam)‡	71.6 / 69.1	71.4 / 66.3	86.6 / 83.9	76.5 / 73.1	Sample	**10.1 sec**	**20**
B^3-Seg (Ours)	**88.3 / 85.4**	**75.3 / 69.7**	**89.8 / 88.0**	**84.5 / 81.0**	**Sample**	**12.1 sec**	**20**

Table 1. **LERF-Mask (accuracy, assumptions, and latency).** Top: Methods that require reconstruction views/labels (=not directly comparable). Bottom: Sampling-based, training-free approach with our 20 views/updates runtime (few seconds). † *Uniform-Sphere*: Candidate viewpoints sampled uniformly on a sphere. ‡ *Recon-Cam*: Candidate viewpoints randomly sampled from reconstruction cameras.

Method	mIoU (%)			
	Bed	Bench	Sofa	Lawn
Assumes reconstruction views/labels (not comparable)				
LangSplat [19]	92.5	94.2	90.0	96.1
Feature 3DGS [32]	83.5	90.7	86.9	93.4
LEGaussians [24]	84.9	91.1	87.8	92.5
Gaussian Grouping [29]	83.0	91.5	87.3	90.6
N2F2 [1]	93.8	92.6	92.1	96.3
SAGA [3]	97.4	95.4	93.5	**96.6**
FastLGS [9]	94.7	95.1	90.6	96.2
LBG [4]	97.7	96.3	**97.3**	87.4
CCL-LGS [27]	97.3	95.0	92.3	96.1
ObjectGS [34]	**98.0**	**96.4**	97.2	95.4
Camera-free & training-free (comparable)				
FlashSplat (Uniform-Sphere)	91.7	86.9	90.2	91.9
FlashSplat (Recon-Cam)	94.3	90.3	85.7	96.3
B^3-Seg (Ours)	**97.1**	**92.2**	**94.1**	**96.8**

Table 2. **3D-OVS (mIoU).** Top: assumes views/labels (*not comparable*). Bottom: camera-free & training-free.

EIG proxy validation. We further verify that our analytic EIG is a faithful proxy for the true information gain. Figure 6 plots the analytic EIG (Eq. (11)) against the measured information gain (Eq. (9)) for selected views in scenes. A strong correlation of $\mathrm{IG}(v)$ and $\mathrm{EIG}(v)$ supports the use of Eq. (11) as a practical ranking surrogate.

Information efficiency over iterations. Figure 7 visualizes the total Beta entropy $\sum_i H(\mathrm{Beta}(a_i, b_i))$ in iterations on the LERF-Mask Teatime scene. Lower entropy indicates higher segmentation certainty. Our EIG-driven policy achieves the largest per-step entropy drop and converges faster than spherical and reconstruction-view sampling, confirming improved information-efficiency.

Runtime analysis. Table 4 reports the runtime breakdown of our method. Mask inference is the dominant cost (9.76 s out of 12.11 s). Our EIG-based view selection only requires lightweight rendering and entropy evaluation, avoiding mask inference on each candidate view.

CLIP re-rank	SAM2 mask-input	Accuracy		Δ vs. Base	
		mIoU	mBIoU	mIoU	mBIoU
–	–	74.9	70.2	–	–
✓	–	81.5	76.6	+6.6	+6.4
✓	✓	84.5	81.0	+9.6	+10.8

Table 3. **Ablation study on LERF-Mask.** Both CLIP re-ranking and SAM2 mask-input improve segmentation performance.

5.4. Ablation Studies

CLIP re-ranking and SAM2 mask-input We evaluated the effectiveness of (i) CLIP re-ranking of candidate masks and (ii) SAM2 with prior mask input. Integrating these improves segmentation quality (Table 3). CLIP filters out inconsistent candidates and SAM2 stabilizes masks. These enhancements complement the EIG-based selection.

Sensitivity analysis. We evaluate the influence of N_{cand} and T in Table 5. Accuracy improves as these parameters increase but quickly saturates around N_{cand}=20 and

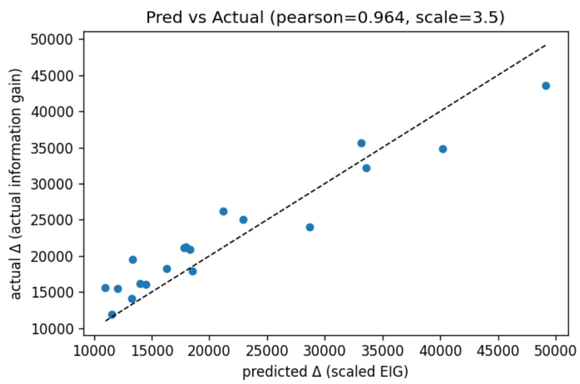

Figure 6. Predicted EIG closely matches information gain on the LERF-Mask, with a strong correlation ($r = 0.964$).

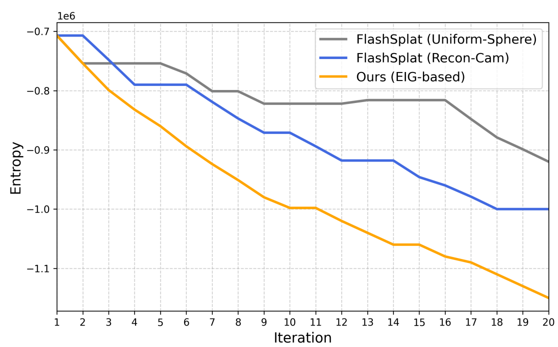

Figure 7. Posterior entropy vs. iteration for three view-selection strategies on the LERF-Mask Teatime scene. EIG-based selection consistently achieves the largest per-step entropy drop.

	Mask Inference	View Selection	Beta Update	Other	Total
Time (s)	9.761	2.111	0.119	0.117	12.108

Table 4. **Runtime breakdown of B^3-Seg**. End-to-end inference completes in 12 seconds for 20 actively selected views.

T=20, showing that our lightweight setting is already sufficient. Furthermore, as shown in Table 6, B^3-Seg remains robust even when the initial object center is perturbed: a 50% shift in $\mathbf{c}_{\mathrm{obj}}$ results in only a 1.6% drop in mIoU. This illustrates that the performance of B^3-Seg is not dependent on its initial condition; EIG-based active view selection quickly compensates for any suboptimal initial condition.

5.5. Discussion and Limitations

B^3-Seg offers a fast and theoretically grounded framework for camera-free 3DGS segmentation. Our experiments focus on typical object-centric scenes, where the proposed analytic EIG and sequential Bayesian updates operate efficiently. Extending the method to substantially larger environments—such as wide indoor spaces or outdoor scans—may require broader viewpoint exploration strategies. Integrating methods like RRT-based camera sampling or multi-scale candidate generation is a promising direction that remains compatible with our analytic EIG formulation.

Additionally, the present probabilistic model addresses binary foreground–background decisions. Many real-world applications involve multiple objects or semantic categories. The framework naturally generalizes to a Dirichlet–Categorical model. This enables multi-object segmentation while preserving the pseudo-count interpretation and the theoretical properties of our EIG-driven updates.

N_{cand}	5	10	20	30
mIoU (%)	76.2	83.1	84.5	84.6
time (s)	10.3	11.6	12.1	14.4
T (iterations)	5	10	20	30
mIoU (%)	79.8	82.7	84.5	84.8
time (s)	3.5	6.6	12.1	19.4

Table 5. **Sensitivity to N_{cand} and T on LERF-Mask.** Accuracy saturates around N_{cand}=20 and T=20 while runtime increases almost linearly.

$\mathbf{c}_{\mathrm{obj}}$ shift	0%	20%	50%	70%	100%
mIoU (%)	84.5	83.6	82.9	81.7	80.7
Δ	–	-0.9	-1.6	-2.8	-3.8

Table 6. **Sensitivity to the initial condition on LERF-Mask.** We shift the initial object center $\mathbf{c}_{\mathrm{obj}}$ along a random 3D direction by a percentage of r_{obj}.

6. Conclusion

We introduce B^3-Seg, a camera-free, training-free, open-vocabulary framework for 3DGS segmentation. The method uses sequential Beta–Bernoulli updates for per-Gaussian labeling and selects camera views via analytic EIG. This Bayesian approach ensures adaptive monotonicity and adaptive submodularity, providing a $(1-1/e)$ approximation guarantee. B^3-Seg achieves competitive scores with the latest methods, without using camera trajectories or ground-truth labels. Our Bayesian framework can be generalized to multi-class segmentation with a Dirichlet–Categorical model and scalability for larger or dynamic scenes, all integrable into the current EIG-based pipeline. These are left for future work.

References

[1] Yash Bhalgat, Iro Laina, Joao F Henriques, Andrew Zisserman, and Andrea Vedaldi. N2f2: Hierarchical scene understanding with nested neural feature fields. In *ECCV*, pages 197–214, 2024. 7

[2] Jiazhong Cen, Zanwei Zhou, Jiemin Fang, Wei Shen, Lingxi Xie, Dongsheng Jiang, Xiaopeng Zhang, Qi Tian, et al. Segment anything in 3d with nerfs. In *NeurIPS*, pages 25971–25990, 2023. 7

[3] Jiazhong Cen, Jiemin Fang, Chen Yang, Lingxi Xie, Xiaopeng Zhang, Wei Shen, and Qi Tian. Segment any 3d gaussians. In *AAAI*, pages 1971–1979, 2025. 7

[4] Rohan Chacko, Nicolai Häni, Eldar Khaliullin, Lin Sun, and Douglas Lee. Lifting by gaussians: A simple, fast and flexible method for 3d instance segmentation. In *WACV*, pages 3497–3507, 2025. 7

[5] Liyan Chen, Huangying Zhan, Kevin Chen, Xiangyu Xu, Qingan Yan, Changjiang Cai, and Yi Xu. Activegamer: Active gaussian mapping through efficient rendering. In *CVPR*, pages 16486–16497, 2025. 2

[6] Liyan Chen, Huangying Zhan, Hairong Yin, Yi Xu, and Philippos Mordohai. Understanding while exploring: Semantics-driven active mapping. In *NeurIPS*, 2025. 2

[7] Daniel Golovin and Andreas Krause. Adaptive submodularity: Theory and applications in active learning and stochastic optimization. *Journal of Artificial Intelligence Research*, 42: 427–486, 2011. 5, 2

[8] Jitesh Jain, Jiachen Li, Mang Tik Chiu, Ali Hassani, Nikita Orlov, and Humphrey Shi. Oneformer: One transformer to rule universal image segmentation. In *CVPR*, pages 2989–2998, 2023. 2

[9] Yuzhou Ji, He Zhu, Junshu Tang, Wuyi Liu, Zhizhong Zhang, Xin Tan, and Yuan Xie. Fastlgs: Speeding up language embedded gaussians with feature grid mapping. In *AAAI*, pages 3922–3930, 2025. 7

[10] Wen Jiang, Boshu Lei, and Kostas Daniilidis. Fisherrf: Active view selection and uncertainty quantification for radiance fields using fisher information. In *ECCV*, 2024. 2

[11] Liren Jin, Xingguang Zhong, Yue Pan, Jens Behley, Cyrill Stachniss, and Marija Popović. Activegs: Active scene reconstruction using gaussian splatting. *IEEE Robotics and Automation Letters*, 2025. 2

[12] Bernhard Kerbl, Georgios Kopanas, Thomas Leimkühler, and George Drettakis. 3d gaussian splatting for real-time radiance field rendering. *ACM Trans. Graph.*, 42(4):139–1, 2023. 1, 2

[13] Justin Kerr, Chung Min Kim, Ken Goldberg, Angjoo Kanazawa, and Matthew Tancik. Lerf: Language embedded radiance fields. In *CVPR*, pages 19729–19739, 2023. 2, 7

[14] 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 *NeurIPS*, pages 53433–53456, 2023. 5

[15] Shilong Liu, Zhaoyang Zeng, Tianhe Ren, Feng Li, Hao Zhang, Jie Yang, Qing Jiang, Chunyuan Li, Jianwei Yang, Hang Su, et al. Grounding dino: Marrying dino with grounded pre-training for open-set object detection. In *ECCV*, pages 38–55, 2024. 4

[16] PlayCanvas Ltd. Supersplat: 3d gaussian splat editor, 2025. Software available from PlayCanvas Ltd. 1

[17] Weijie Lyu, Xueting Li, Abhijit Kundu, Yi-Hsuan Tsai, and Ming-Hsuan Yang. Gaga: Group any gaussians via 3d-aware memory bank. In *AAAI*, 2025. 7

[18] Juliette Marrie, Romain Ménégaux, Michael Arbel, Diane Larlus, and Julien Mairal. Ludvig: Learning-free uplifting of 2d visual features to gaussian splatting scenes. In *ICCV*, 2025. 1

[19] Minghan Qin, Wanhua Li, Jiawei Zhou, Haoqian Wang, and Hanspeter Pfister. Langsplat: 3d language gaussian splatting. In *CVPR*, pages 20051–20060, 2024. 1, 2, 7

[20] Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, et al. Learning transferable visual models from natural language supervision. In *ICML*, pages 8748–8763. PmLR, 2021. 2, 5

[21] Nikhila Ravi, Valentin Gabeur, Yuan-Ting Hu, Ronghang Hu, Chaitanya Ryali, Tengyu Ma, Haitham Khedr, Roman Rädle, Chloe Rolland, Laura Gustafson, Eric Mintun, Junting Pan, Kalyan Vasudev Alwala, Nicolas Carion, Chao-Yuan Wu, Ross Girshick, Piotr Dollár, and Christoph Feichtenhofer. Sam 2: Segment anything in images and videos. *arXiv preprint arXiv:2408.00714*, 2024. 4

[22] Tianhe Ren, Shilong Liu, Ailing Zeng, Jing Lin, Kunchang Li, He Cao, Jiayu Chen, Xinyu Huang, Yukang Chen, Feng Yan, et al. Grounded sam: Assembling open-world models for diverse visual tasks. *arXiv preprint arXiv:2401.14159*, 2024. 2

[23] Qiuhong Shen, Xingyi Yang, and Xinchao Wang. Flashsplat: 2d to 3d gaussian splatting segmentation solved optimally. In *ECCV*, pages 456–472, 2024. 1, 2, 3, 7

[24] Jin-Chuan Shi, Miao Wang, Hao-Bin Duan, and Shao-Hua Guan. Language embedded 3d gaussians for open-vocabulary scene understanding. In *CVPR*, pages 5333–5343, 2024. 7

[25] Julian Straub, Thomas Whelan, Lingni Ma, Yufan Chen, Erik Wijmans, Simon Green, Jakob J Engel, Raul Mur-Artal, Carl Ren, Shobhit Verma, et al. The replica dataset: A digital replica of indoor spaces. *arXiv preprint arXiv:1906.05797*, 2019. 2

[26] Matthew Tancik, Ethan Weber, Evonne Ng, Ruilong Li, Brent Yi, Terrance Wang, Alexander Kristoffersen, Jake Austin, Kamyar Salahi, Abhik Ahuja, et al. Nerfstudio: A modular framework for neural radiance field development. In *SIGGRAPH*, pages 1–12, 2023. 1

[27] Lei Tian, Xiaomin Li, Liqian Ma, Hao Yin, Zirui Zheng, Hefei Huang, Taiqing Li, Huchuan Lu, and Xu Jia. Ccl-lgs: Contrastive codebook learning for 3d language gaussian splatting. In *ICCV*, pages 9855–9864, 2025. 7

[28] Yanmin Wu, Jiarui Meng, Haijie Li, Chenming Wu, Yahao Shi, Xinhua Cheng, Chen Zhao, Haocheng Feng, Errui Ding, Jingdong Wang, et al. Opengaussian: Towards point-level 3d gaussian-based open vocabulary understanding. In *NeurIPS*, pages 19114–19138, 2024. 1, 2

[29] Mingqiao Ye, Martin Danelljan, Fisher Yu, and Lei Ke. Gaussian grouping: Segment and edit anything in 3d scenes. In *ECCV*, pages 162–179, 2024. 1, 2, 5, 7

[30] Jiaxin Zhang, Junjun Jiang, Youyu Chen, Kui Jiang, and Xianming Liu. Cob-gs: Clear object boundaries in 3dgs segmentation based on boundary-adaptive gaussian splitting. In *CVPR*, pages 19335–19344, 2025. 1, 2, 3

[31] Yian Zhao, Wanshi Xu, Ruochong Zheng, Pengchong Qiao, Chang Liu, and Jie Chen. isegman: Interactive segment-and-manipulate 3d gaussians. In *CVPR*, pages 661–670, 2025. 2

[32] Shijie Zhou, Haoran Chang, Sicheng Jiang, Zhiwen Fan, Zehao Zhu, Dejia Xu, Pradyumna Chari, Suya You, Zhangyang Wang, and Achuta Kadambi. Feature 3dgs: Supercharging 3d gaussian splatting to enable distilled feature fields. In *CVPR*, pages 21676–21685, 2024. 7

[33] Runsong Zhu, Shi Qiu, Zhengzhe Liu, Ka-Hei Hui, Qianyi Wu, Pheng-Ann Heng, and Chi-Wing Fu. Rethinking end-to-end 2d to 3d scene segmentation in gaussian splatting. In *CVPR*, pages 3656–3665, 2025. 7

[34] Ruijie Zhu, Mulin Yu, Linning Xu, Lihan Jiang, Yixuan Li, Tianzhu Zhang, Jiangmiao Pang, and Bo Dai. Objectgs: Object-aware scene reconstruction and scene understanding via gaussian splatting. In *ICCV*, 2025. 1, 2, 5, 7