CVPR 2026 HIGHLIGHT

B³-Seg

Camera-Free, Training-Free Open-Vocabulary 3DGS Segmentation

Hiromichi Kamata[1] Samuel Arthur Munro[2] Fuminori Homma[1]

[1]Sony Group Corporation [2]Pixomondo

arXiv: 2602.17134 Project: sony.github.io/B3-Seg-project/

SECTION 02 · INTRODUCTION

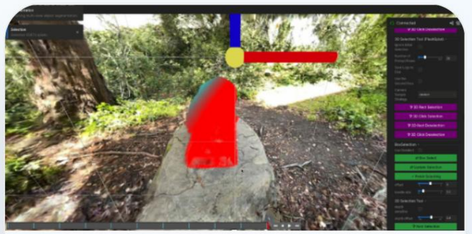
3DGS Segmentation: Labeling Objects in 3D Gaussian Scenes

3D Gaussian Splatting (3DGS) represents scenes as millions of 3D Gaussians for real-time novel-view synthesis. **3DGS segmentation** assigns each Gaussian to an object, making the scene *queryable and editable*.

What does 3DGS segmentation enable?

- Interactive object extraction
- Edit, remove, or replace objects
- Open-vocabulary queries
- Robotics, VFX, digital twins

User interaction: click, drag, or text input to select objects



Open-vocabulary object extraction

Unlike 2D segmentation, labels must stay **consistent across all viewpoints** — one per Gaussian.

SECTION 03 · MOTIVATION

Why Fast, Interactive 3DGS Segmentation Matters

🎬 VFX & Film Production

Artists use 3DGS scenes to **isolate props, characters, and backgrounds** for compositing.

Object Isolation · Scene Editing · Compositing

🤖 Robotics & Scene Understanding

Robots must **localize and segment target objects** directly in the 3D map.

Object Localization · Manipulation · Scene Maps

🥽 AR/VR & Digital Twins

Specific components must be segmented on-the-fly in AR/VR and digital twins.

Digital Twins · AR Overlay · Object Tracking

Common bottleneck: current methods need cameras or manual labels, so we need **camera-free, training-free** segmentation.

SECTION 04 · PROBLEM SETTING

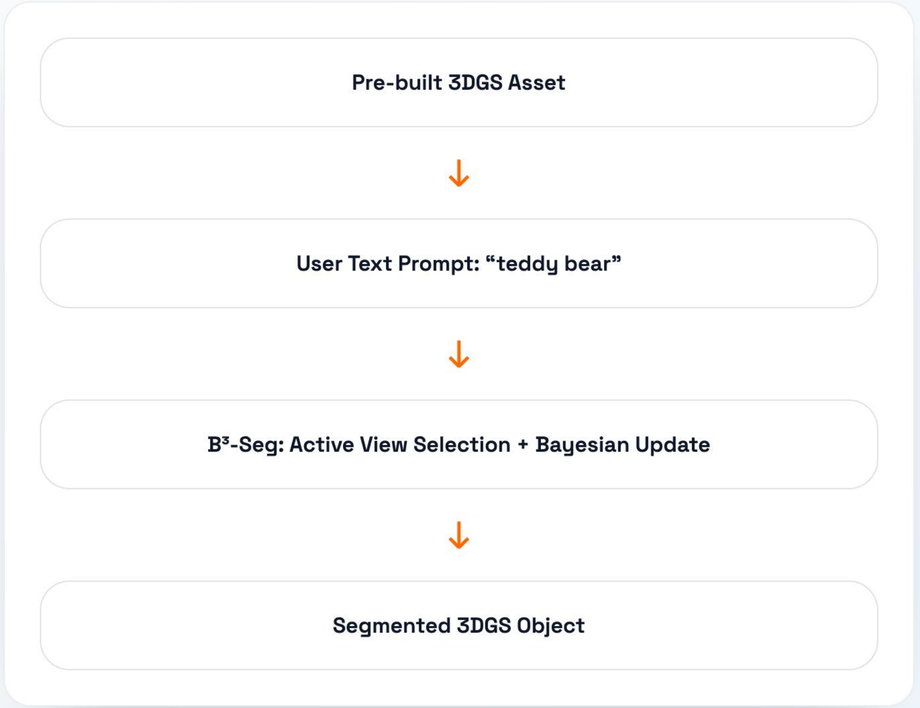
The Problem: Interactive 3DGS Segmentation

Camera-Free
Use only the pre-built 3DGS asset.

Training-Free
Work out of the box with no scene-specific training.

Open-Vocabulary
Specify targets with text, e.g., “the red chair”.

Few Seconds
Interactive use needs seconds, not minutes.



This is unlike the standard segmentation setting: we only have the 3DGS scene and a text query.

SECTION 05 · CHALLENGE

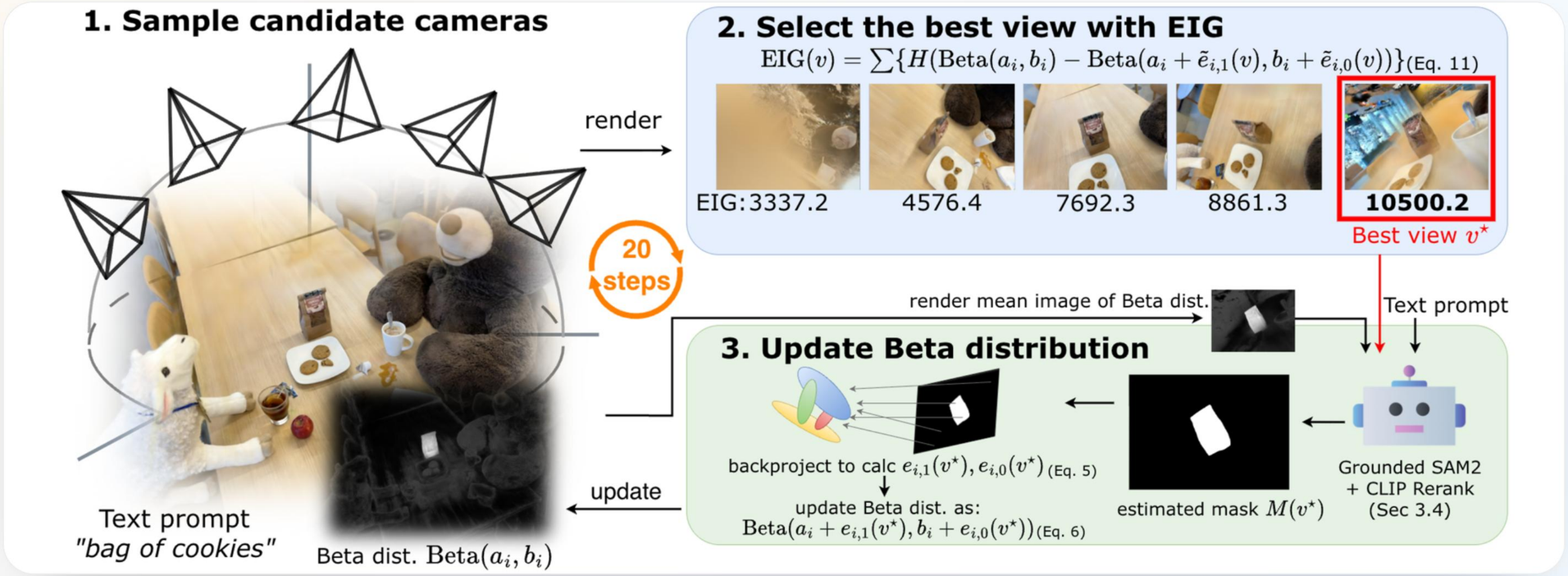
Why This Is Hard: Existing Methods Fall Short

3DGS is *view-dependent*, so object identity depends on which viewpoints you inspect. Prior methods need reconstruction cameras or exhaustive view search, which is unavailable or too slow.

Method	Requires Recon Cameras	Training?	Camera-Free?
LERF	✗ Yes	✗ Per-scene	✗
Gaussian Grouping	✗ Yes	✗ Per-scene	✗
OpenGaussian	✗ Yes	✗ Per-scene	✗
FlashSplat	✗ Yes	✓ No	✗
COB-GS	✗ Yes	✗ Per-scene	✗
B³-Seg (Ours)	**✓ No**	**✓ No**	**✓ Yes**

SECTION 06 · PIPELINE

Our Solution: B³-Seg Pipeline



① **Initialize**
Set Beta(1,1) priors for each Gaussian.

② **Sample Candidates**
Sample camera-free views on a sphere.

③ **Score by EIG**
Score each view with analytic EIG from one render.

④ **Select + Update**
Run SAM2 + CLIP once, then update the Beta posterior.

Key innovation: **We never run SAM2 on all candidates.** Analytic EIG picks the best view first.

SECTION 07 · BAYESIAN REFORMULATION

Method: Bayesian Reformulation of 3DGS Segmentation

FlashSplat (Prior Work)

Deterministic binary assignment via linear program

$$P_i = \arg\max_n \sum_v \alpha_i T_i \mathbb{1}[M(v) = n]$$

- ✗ No uncertainty model
- ✗ Cannot reason about which view to pick next

B³-Seg (Ours)

Sequential Bayesian inference with Beta-Bernoulli model

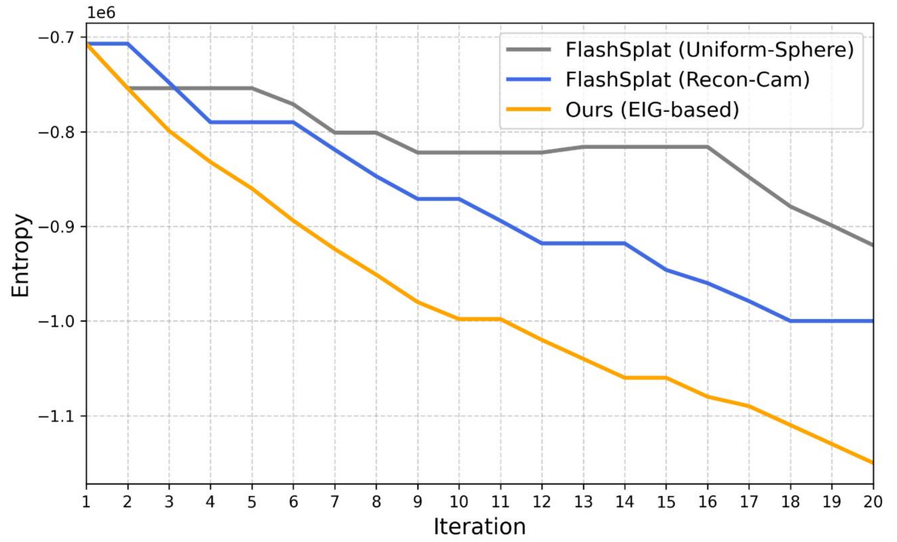
$$p_i \sim \mathrm{Beta}(a_i, b_i), \quad y_i \mid p_i \sim \mathrm{Bernoulli}(p_i)$$

$$\mathrm{Beta}(a_i, b_i) \xrightarrow{\text{view } v} \mathrm{Beta}(a_i + e_{i,1}, b_i + e_{i,0})$$

- ✓ Per-Gaussian uncertainty quantification
- ✓ Enables EIG-based view planning

Key Insight

Why Beta-Bernoulli? — Conjugacy gives a **closed-form posterior update**. No MCMC, no variational inference. Each new view reduces Beta entropy, and this entropy reduction *is* the information gain we want to maximize.



Beta entropy decreases with each new view — driving EIG-guided selection toward more informative viewpoints.

SECTION 08 · ANALYTIC EIG

Method: Analytic EIG for Efficient View Selection

Naive approach: Score each candidate view by running the full pipeline (SAM2 + CLIP + Beta update). Cost: O(N × SAM2). With 50 candidates, this means 50 full SAM2 inferences just to pick *one* view. Too slow.

Analytic EIG: Approximate the expected evidence using current Beta means. Only 1 render per candidate — SAM2 runs only on the *selected* view.

Step 1 — Approximate Evidence

$$\tilde{e}_{i,1}(v) = m_i \cdot \tau_i, \quad \tilde{e}_{i,0}(v) = (1 - m_i) \cdot \tau_i$$

where $m_i = a_i/(a_i + b_i)$ and τ_i is the total transmittance from the rasterizer.

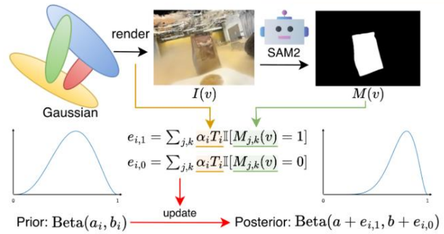
Step 2 — Compute EIG

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

Step 3 — Select Best

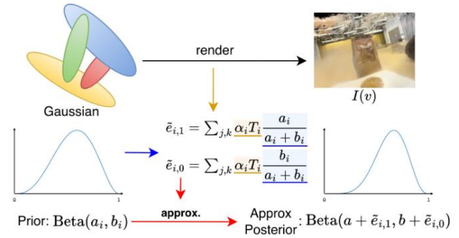
$$v^* = \arg\max_{v \in \mathcal{V}} \mathrm{EIG}(v)$$

Greedy selection naturally concentrates compute on views that directly resolve object uncertainty.



$$\mathrm{IG} = H(\mathrm{Beta}(a, b)) - H(\mathrm{Beta}(a + e_1, b + e_0))$$

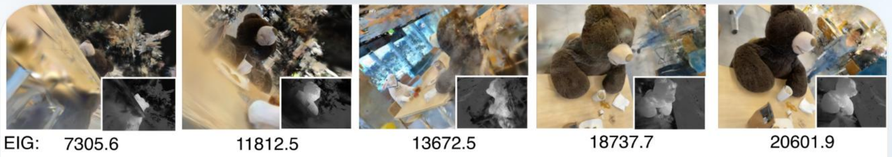
(a) Information Gain calculation



$$\mathrm{EIG} = H(\mathrm{Beta}(a, b)) - H(\mathrm{Beta}(a + \tilde{e}_1, b + \tilde{e}_0))$$

(b) Expected Information Gain calculation

Analytic EIG eliminates the need for per-candidate SAM2 inference — only the selected best view requires segmentation.



Views with higher EIG tend to have less occlusion, offering a clearer look at the target object.

SECTION 09 · THEORY

Theoretical Guarantees: Near-Optimal by Design

Greedy EIG selection is not just a heuristic — it has provable optimality guarantees.

LEMMA 1

Adaptive Monotonicity

$\mathrm{EIG}(v \mid S) \geq 0$ for any view v and any set S of already-selected views.

Every new view reduces uncertainty — no view is ever "useless". Viewing from any direction gives us at least some information about which Gaussians belong to the object.

LEMMA 2

Adaptive Submodularity

$\mathrm{EIG}(v \mid S) \geq \mathrm{EIG}(v \mid S')$ for $S' \supseteq S$.

Diminishing returns: the more views you've already taken, the less new information each additional view provides. This is the key property that enables efficient greedy selection.

THEOREM

Greedy (1−1/e) Approximation

(1−1/e) ≈ 63% of Optimal

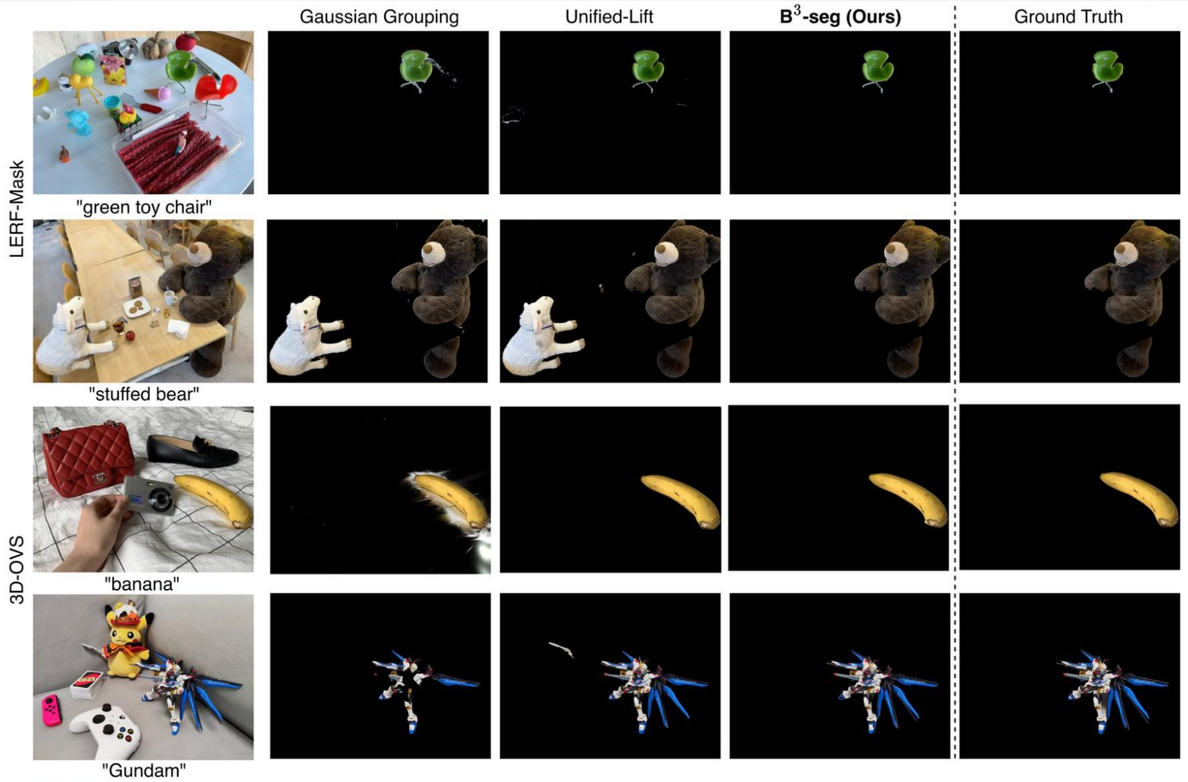
$$\mathbb{E}[\mathrm{EIG}(S_k^{\mathrm{greedy}})] \geq \left(1 - \frac{1}{e}\right) \cdot \max_{\pi} \mathbb{E}[\mathrm{EIG}(S_k^{\pi})]$$

Greedy selection achieves at least 63% of the best possible adaptive policy — guaranteed, regardless of the scene.

These guarantees hold under the Beta-Bernoulli model assumptions. The key technical contribution is proving that the **analytic EIG approximation** (which uses a single render) preserves these properties from the exact EIG.

SECTION 10 · QUALITATIVE RESULTS

Qualitative Results: Clean 3D Segmentation Masks



Comparison on LERF-Mask benchmark. B³-Seg produces consistently cleaner segmentation boundaries and more complete masks, especially for objects with complex geometry.

✓ Cleaner object boundaries

✓ More complete masks

✓ Fewer false positives

SECTION 11 · QUANTITATIVE RESULTS

Quantitative Results: Competitive with Supervised Methods

84.5
mIoU ↑
LERF-Mask
+14.9 vs baseline

95.1
mIoU ↑
3D-OVS
+4.9 vs baseline

12.1 sec
End-to-End
RTX A6000
vs. ~minutes

LERF-Mask Benchmark

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	∼ 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.

3D-OVS Benchmark

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.

SECTION 12 · SUMMARY

Summary

Problem

Interactive 3DGS segmentation needs camera-free, training-free, open-vocabulary operation in seconds.

Method

Beta-Bernoulli updates track uncertainty, and **analytic EIG** picks views before running SAM2.

Theory

EIG is adaptive monotone + submodular, so greedy achieves **(1−1/e) ≈ 63% of optimal**.

Results

84.5 / 95.1 mIoU on LERF-Mask / 3D-OVS in **12.1 seconds** on RTX A6000.

arXiv: 2602.17134
Project: sony.github.io/B3-Seg-project/

```
@inproceedings{kamata2026b3seg,
  title={B$^3$-Seg: Camera-Free, Training-Free 3DGS Segmentation via Analytic EIG and Beta-Bernoulli Bayesian Updates},
  author={Hiromichi Kamata and Samuel Arthur Munro and Fuminori Homma},
  booktitle={Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)},
  year={2026},
  url={https://arxiv.org/abs/2602.17134}
}
```