NG-GS: NeRF-Guided 3D Gaussian Splatting Segmentation

Supplementary Material

The supplementary material provides complete implementation details and extended analyses. Specifically, it includes: the Edge Gaussian Continuity algorithm (Sec. 7), a computation cost analysis comparing efficiency (Sec. 8), an open-vocabulary segmentation evaluation (Sec. 9, Table 8), a test of robustness to erroneous masks (Sec. 10), and a hyperparameter analysis (Sec. 11). We provide expanded quantitative results: a computation cost comparison (Table 7), per-scene results on LERF-OVS (Table 8), and CLIP-IQA scores (Table 9).

7. Edge Gaussian Continuity

The Edge Gaussian Continuity algorithm is designed to address the discrete boundary issues in 3DGS by constructing a spatially continuous feature field. It begins by identifying boundary Gaussians through variance analysis of multi-view 2D masks, where Gaussians with high variance values are selected as ambiguous boundary points. These points are then projected onto a reference image plane to compute an expanded bounding box. Dense grid sampling is performed within this box to generate 2D sample points, which are subsequently used to create 3D query points along camera rays via depth-adaptive sampling. For each query point, RBF interpolation is applied using K-nearest neighbors from the boundary Gaussians, with weights based on Gaussian kernel distances. Finally, MRHE enhances the interpolated features by capturing multi-scale spatial information, which improves boundary smoothness and segmentation consistency in the NG-GS framework. See Algorithm 1 for details.

8. Computation Cost

We compare the computational efficiency of NG-GS with state-of-the-art 3DGS based methods and feedforward based methods. This evaluation was conducted using a single NVIDIA RTX 3090 GPU on all scenarios from the NOVS dataset [30], and the results are shown in Table 7. We provide the average total training time and inference time for the entire reconstruction and segmentation pipeline. Compared with COB-GS, our segmentation process does not rely on edge Gaussian splitting to remove mutated Gaussian, but utilizes NeRF and fast modeling MRHE, which ensures edge optimization while optimizing scene labels. The optimization time is comparable to the speed of COB-GS.

Algorithm 1 Edge Gaussian Continuity

Input: Trained 3DGS model $\mathcal{G}$, 2D mask set M, camera parameters, variance threshold τ, grid size N_{row}, N_{col}, sampling depth K, kernel width σ, expansion offset δ
Result: Spatially continuous feature field
// Step 1: Identify boundary Gaussians
for each Gaussian g_i in $\mathcal{G}$ **do**
  $v_i \leftarrow \text{ComputeMaskVariance}(M, g_i)$
  if $v_i > \tau$ **then**
    $\mathcal{B}[g_i] \leftarrow g_i$
  end if
end for
// Step 2: Project and compute bounding box
Project all $\mathcal{B}$ to image coordinates for a reference view
$B_{min} \leftarrow \min(\text{coordinates}) - \delta$
$B_{max} \leftarrow \max(\text{coordinates}) + \delta$
// Step 3: Dense grid sampling
$\Delta x \leftarrow (B_{max}.x - B_{min}.x)/N_{col}$
$\Delta y \leftarrow (B_{max}.y - B_{min}.y)/N_{row}$
for $k \leftarrow 0$ to N_{col} **do**
  for $l \leftarrow 0$ to N_{row} **do**
    $p \leftarrow (B_{min}.x + k \cdot \Delta x, B_{min}.y + l \cdot \Delta y)$
    $\mathcal{P}_{grid} \leftarrow \mathcal{P}_{grid} \cup \{p\}$
  end for
end for
// Step 4: Generate query points along rays
for each sample point p_i in $\mathcal{P}_{grid}$ **do**
  $d_{world} \leftarrow \text{ComputeRayDirection}(p_i, \text{camera params})$
  for $s \leftarrow 1$ to K **do**
    $\delta_s \leftarrow -1 + 2 \cdot (s-1)/(K-1)$
    $q \leftarrow o + [\bar{t} + \max(\alpha \cdot \bar{t}, \varepsilon) \cdot \delta_s] \cdot d_{world}$
    $\mathcal{P}_{query} \leftarrow \mathcal{P}_{query} \cup \{q\}$
  end for
end for
// Step 5: RBF interpolation
for each query point q in $\mathcal{P}_{query}$ **do**
  $\mathcal{N}_q \leftarrow \text{KNN}(\mathcal{B}, q)$
  for each Gaussian g_j in $\mathcal{N}_q$ **do**
    $w_j \leftarrow \exp(-\|q - x_j\|^2/(2\sigma^2))$
  end for
  Normalize weights w_j to sum to 1
  $f^{inter}(q) \leftarrow \sum_{g_j \in \mathcal{N}_q} w_j \cdot f_j$
end for
// Step 6: Multi-resolution hash encoding
for each query point q in $\mathcal{P}_{query}$ **do**
  $f^{hash} \leftarrow \text{MRHE}(q)$
end for

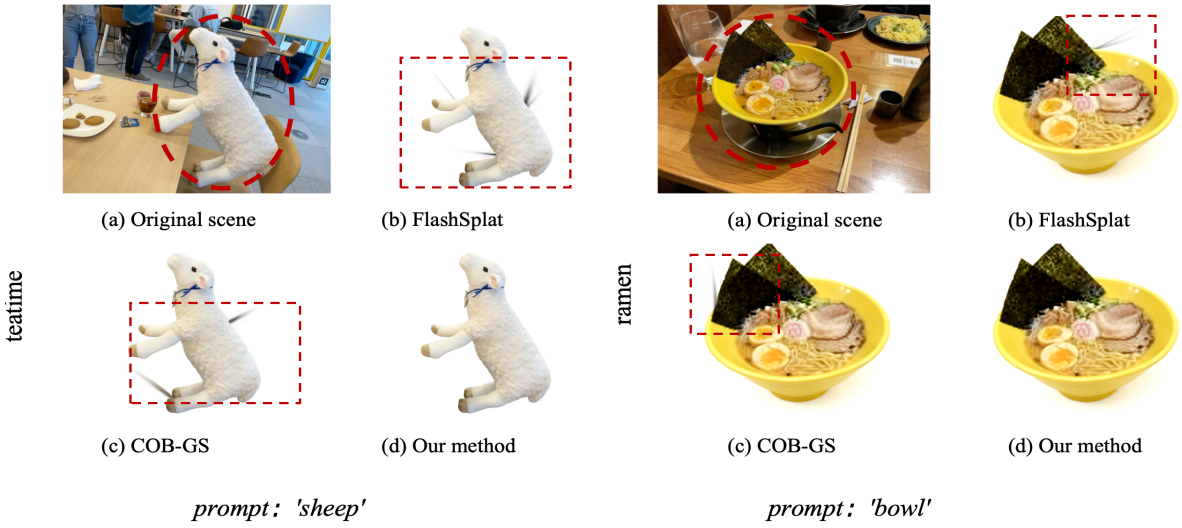


Figure 6. More qualitative comparisons of open-vocabulary 3D Gaussian Splatting segmentation on LERF-OVS dataset.

Table 7. Average training and inference times consumption comparison across all scenes.

Metric	LSeg	LangSplat	LangSurf	LSM	LanScene-X	SA3D-GS	SAGA	FlashSplat	COB-GS	Ours
Training time (m)	–	–	–	–	–	17	14	15	13	**12**
Inference time (s)	1.8	1.5	1.3	1.4	1.2	1.4	1.2	1.1	1.0	**0.9**

Table 8. Quantitative results on LERF-OVS dataset comparing various methods across three different scenes.

Method	**Figurines**		**Ramen**		**Teatime**	
	B-mIoU (%)	mIoU (%)	B-mIoU (%)	mIoU (%)	B-mIoU (%)	mIoU (%)
LSeg [16]	45.2	48.3	42.8	51.1	25.3	30.5
LangSplat [27]	11.9	13.8	10.2	15.0	12.3	15.8
LangSurf [17]	15.2	18.9	13.4	21.7	14.7	18.8
LSM [9]	18.1	21.1	14.8	23.0	13.8	19.6
LanScene-X [19]	37.4	40.1	34.7	42.9	40.1	45.0
SA3D-GS [3]	23.9	24.9	7.0	7.4	39.2	42.5
SAGA [2]	45.2	48.5	37.2	41.2	55.3	60.1
FlashSplat [33]	50.5	52.8	44.7	50.4	65.6	69.5
COB-GS [37]	73.9	76.3	69.2	78.1	72.8	77.2
Ours	**74.2**	**75.4**	**77.5**	**79.3**	**76.3**	**79.2**

Table 9. Quantitative comparison of CLIP-IQA for mask-based 3DGS segmentation methods.

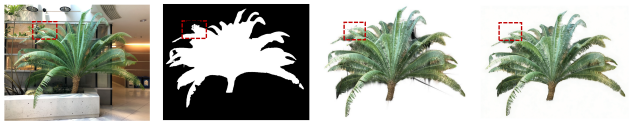
Method	**CLIP-IQA**		
	Clear / Unclear Boundary	Smooth / Noisy Boundary	Complete / Mutilated Object
SA3D-GS [3]	65.8	71.8	83.5
SAGA [2]	66.2	62.4	84.6
FlashSplat [33]	62.6	64.4	82.9
COB-GS [37]	68.2	73.1	85.9
Ours	**72.6**	**75.4**	**88.3**

9. Open-Vocabulary 3D Segmentation

We use the COB-GS to implement open vocabulary semantic segmentation. Some objects in this dataset have severe occlusion, and we show more examples of open vocabulary 3D semantic segmentation on the LERF-OVS [14] dataset in Figure 6. We observed that the results generated by COB-GS cannot provide the exact shape of the query object and contain a lot of noise, while our method can accurately describe the shape of the object. These results demonstrate the effectiveness of our proposed NG-GS in object edge segmentation as shown in Table 8. The CLIP-IQA test compared with COB-GS is provided in the Table 9.

10. Robustness Against Erroneous Masks

Our method inherently addresses this concern via $\mathcal{L}_{\text{smth}}$ and RBF interpolation, and achieves more accurate results than COB-GS under the erroneous masks, showing in the Figure 7.



(a) Input (b) GT (c) COB-GS (d) Our method

Figure 7. Illustration robustness against erroneous masks.

11. Hyperparameter Experiment

As shown in Figure 8, the parameter σ remained stable within $[0.3, 0.4]$, and $K = 8$ was determined via grid search to effectively balance underfitting and noise.

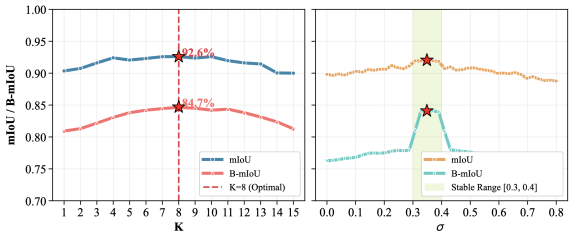


Figure 8. The impact of σ and K on mIoU and B-mIoU.