COB-GS: Clear Object Boundaries in 3DGS Segmentation Based on Boundary-Adaptive Gaussian Splitting

Supplementary Material

In the supplementary material, we first introduce in detail our proposed two-stage mask generation based on text prompts in Sec. 6. Next, we present the concrete training strategy and implementation details of COB-GS in Sec. 7. In Sec. 8 and Sec. 9, we evaluate the open-vocabulary segmentation capability and the computational cost of COB-GS, respectively. Finally, additional visualizations of the segmentation results are presented in Sec. 10.

6. Two-Stage Mask Generation Based on Text Prompts

Mask-based 3D segmentation requires generating a set of masks for regions of interest from a collection of input images. Thus, the supervision data consists of V input views $\{I^v\}$ corresponding to 2D binary masks $\{M^v\}$. Each mask $M \in \mathbb{R}^{H \times W}$ contains discrete values of 0 and 1. The related work SA3D [4] improves optimization efficiency and mask view consistency by using Segment Anything Model (SAM) [9] to iteratively generate the mask for each frame. With the emergence of foundational models like SAM2 [13], mask prediction across video sequences has become feasible.

SAM2 retains the encoder-decoder structure of SAM, where the encoder S_e takes an image I as input. Unlike SAM, SAM2 employs memory attention S_m to tilize past frame features f_m as conditions for generating the current frame embedding e_I:

$$e_I = S_m(S_e(I), f_m) \tag{1}$$

The past frame features f_m are maintained in a FIFO memory queue. The decoder takes the current frame embedding e_I and the prompts P as input, outputting the corresponding 2D binary mask M:

$$M = S_d(e_I, P) \tag{2}$$

The prompts P include masks, boxes, points, or texts. The memory capability of SAM2 allows it to handle mask prediction for video sequences, which aligns with the input view conditions $\{I^v\}$ for the 3DGS task. However, when SAM2 performs mask prediction across video sequences, it encounters challenges with object continuity; specifically, it may fail to recognize severely occluded objects due to information discontinuity.To address this issue, we propose a two-stage mask generation method based on text prompts. In the coarse mask generation stage, we utilize Grounding DINO [2] to extract box prompts from the given prompt frame with lower text confidence, which are then used for full-sequence mask prediction to obtain preliminary results. In the fine-grained stage, we leverage Grounding DINO with higher text confidence to extract box prompts for subsequences within the original sequence that lack mask prediction results, which are then used for subsequence mask prediction. See Algorithm 1 for details.

Algorithm 1 Two-stage mask generation

Input: Frame index idx, text prompt $text$, image set I, high confidence C_{high}, low confidence C_{low}
Result: Updated dictionary $video\_segments$
Initialize dictionary $valid\_idxs \leftarrow \{\}$
Initialize dictionary $video\_segments \leftarrow \{\}$
SAM2.init_state(I)
$image \leftarrow I[idx]$
$boxes \leftarrow$ Grounding DINO($text, image, C_{low}$)
SAM2.add_new_box($idx, boxes$)
for each frame $i, mask$ in SAM2(idx) **do**
  $video\_segments[i] \leftarrow mask$
  $valid\_idxs[i] \leftarrow$ if $mask$ is empty then 0 else 1
end for
for each key in $valid\_idxs$ **do**
  if $valid\_idxs[key] = 0$ **then**
    $boxes \leftarrow$ Grounding DINO($text, I[key], C_{high}$)
    if boxes is empty **then**
      continue
    end if
    SAM2.add_new_box($idx, boxes$)
    $max\_sk \leftarrow$ FindMaxSub($valid\_idxs, key$)
    for each frame $j, mask$ in SAM2(key, max_sk) **do**
      $video\_segments[j] \leftarrow mask$
      $valid\_idxs[j] \leftarrow$ if $mask$ is empty then 0 else 1
    end for
  end if
end for

7. Implementation Details

Our method is a post-processing method based on the original 3D Gaussian Splatting [7]. For each scene, we perform 30,000 iterations of training according to the parameters set by the original 3DGS to obtain the original 3DGS scene. COB-GS mainly consists of two components: optimization process and robustness process. The optimization process involves alternating between mask optimization and texture optimization. For the mask optimization stage, we optimize

Table 4. Results on LERF-mask dataset.

Method	Figurines		Ramen		Teatime	
	mIoU (%)	mBIoU (%)	mIoU (%)	mBIoU (%)	mIoU (%)	mBIoU (%)
DEVA [5]	46.2	45.1	56.8	51.1	54.3	52.2
LERF [8]	33.5	30.6	28.3	14.7	49.7	42.6
SA3D [4]	24.9	23.9	7.4	7.0	42.5	39.2
LangSplat [12]	52.8	50.5	50.4	44.7	69.5	65.6
Gaussian Grouping [15]	69.7	67.9	77.0	68.7	71.7	66.1
COB-GS (ours)	**76.3**	**73.9**	**78.1**	**69.2**	**77.2**	**72.8**

Figure 9. Visualization of the LERF-mask dataset [15]. The result of the segmentation is obtained under the specified text prompt.

the mask labels and perform Gaussian splitting. The learning rate of the mask labels is set to 0.1. For the texture optimization stage, we optimize the geometry and texture, and the learning rate of appearance follows the original 3DGS setting. Each stage is trained for $2\times V$ iterations, where V is the number of input images. Two sets of hyperparameters are used for different scene types: for forward scenes, we set $\delta = 0.5$ and perform a total of $22\times V$ iterations of alternating optimization; for surrounding scenes, we set $\delta = 0.8$ and conduct $14\times V$ iterations of alternating optimization. The robustness process follows scene optimization and involves extracting and refining ambiguous boundary Gaussians at scales smaller than the pixel scale. In our two-stage mask generation method, we utilize the SAM2 hiera_l model and the Grounding DINO swinb model. All experiments were conducted on a single NVIDIA RTX 3090 GPU.

8. Open-Vocabulary 3D Segmentation

To achieve open-vocabulary semantic segmentation, we follow the setup of existing methods [4, 15] and utilize Grounding DINO [2] to generate boxes for input images, similar to the approach in Sec. 6. We compare our method with the current state-of-the-art methods for open-vocabulary 3D segmentation using the LERF-mask dataset, which is annotated from test views of three 3D scenes in the LERF dataset. The scenes contain severe object occlusions, and the mask boundaries of the test views are more complex. As shown in Table 4, our method demonstrates a clear advantage over

current SOTA methods. Visual segmentation comparisons in Figure 9 reveal that our method provides more accurate segmentation predictions with clear boundaries, while Gaussian Grouping [15] exhibits blurriness in segmentation results.

9. Computation Cost

We evaluate the computational efficiency of COB-GS in comparison to state-of-the-art 3DGS segmentation methods, namely the feature-based SAGA [3] and the mask-based FlashSplat [14]. This evaluation is conducted on the Fortress scene ($V = 42$) from the LLFF dataset [10] using a single NVIDIA RTX 3090 GPU, with results presented in Table 5. We provide the total time cost (prep time+opt time+seg time) and the maximum VRAM of the entire reconstruction and segmentation pipeline. SAGA [3] requires 10,000 iterations of gradient descent to distill 2D masks into object features associated with each 3D Gaussian, resulting in substantial additional training time for scene optimization. Moreover, object segmentation remains time-consuming due to the need for network inference. FlashSplat [14] does not offer a mask extraction method, and assigning labels to each Gaussian through forward rendering is relatively time-consuming. In contrast, our extraction process relies entirely on inverse rendering, which ensures that texture optimization simultaneously optimizes scene labels. The optimization time is comparable to the speed of FlashSplat, and segmentation requires only filtering the labels.

Method	Prep Time	Opt Time	Seg Time	Total Time	Mem
SAGA [3]	145 s	20 min	200 ms	22.42 min	7.6 G
FlashSplat [14]	N/A	24 s	10 ms	N/A	2.4 G
COB-GS	4 s	24 s	8 ms	0.46 min	2.7 G

Table 5. Computation cost comparisons over the Fortress scene.

10. More Qualitative Results

To demonstrate the effectiveness of our proposed 3D segmentation method in producing clear object boundaries, we provide visualizations of 3D segmentation across multiple scenes, including the Horns, Orchids and Fortress from the LLFF dataset [10], the Garden from MIP-360 [1], the Bear from the IN2N dataset [6], and the Pinecone from NeRF [11], encompassing both forward and surrounding scenes. We obtain masks using text prompts, as described in Sec. 6. The results shown in Figure 10 clearly demonstrate that the object edges in our 3D segmentation results are very clear, while also maintaining high-quality textures for both the foreground and background.

References

[1] Jonathan T. Barron, Ben Mildenhall, Matthew Tancik, Peter Hedman, Ricardo Martin-Brualla, and Pratul P. Srinivasan. Mip-NeRF: A Multiscale Representation for Anti-Aliasing Neural Radiance Fields. In *ICCV*, pages 5855–5864, 2021. 3

[2] Mathilde Caron, Hugo Touvron, Ishan Misra, Hervé Jégou, Julien Mairal, Piotr Bojanowski, and Armand Joulin. Emerging Properties in Self-Supervised Vision Transformers. In *ICCV*, pages 9650–9660, 2021. 1, 2

[3] Jiazhong Cen, Jiemin Fang, Chen Yang, Lingxi Xie, Xiaopeng Zhang, Wei Shen, and Qi Tian. Segment Any 3D Gaussians. *arXiv preprint arXiv:2312.00860*, 2023. 3

[4] Jiazhong Cen, Zanwei Zhou, Jiemin Fang, Chen Yang, Wei Shen, Lingxi Xie, Dongsheng Jiang, Xiaopeng Zhang, and Qi Tian. Segment Anything in 3D with NeRFs. *NeurIPS*, 36: 25971–25990, 2023. 1, 2

[5] Ho Kei Cheng, Seoung Wug Oh, Brian Price, Alexander Schwing, and Joon-Young Lee. Tracking Anything with Decoupled Video Segmentation. In *ICCV*, pages 1316–1326, 2023. 2

[6] Ayaan Haque, Matthew Tancik, Alexei Efros, Aleksander Holynski, and Angjoo Kanazawa. Instruct-NeRF2NeRF: Editing 3D Scenes with Instructions. In *ICCV*, pages 19740–19750, 2023. 3

[7] Bernhard Kerbl, Georgios Kopanas, Thomas Leimkühler, and George Drettakis. 3d gaussian splatting for real-time radiance field rendering. *TOG*, 42(4):1–14, 2023. 1

[8] Justin Kerr, Chung Min Kim, Ken Goldberg, Angjoo Kanazawa, and Matthew Tancik. LERF: Language Embedded Radiance Fields. In *ICCV*, pages 19729–19739, 2023. 2

[9] 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. *arXiv preprint arXiv:2304.02643*, 2023. 1

[10] Ben Mildenhall, Pratul P. Srinivasan, Rodrigo Ortiz-Cayon, Nima Khademi Kalantari, Ravi Ramamoorthi, Ren Ng, and Abhishek Kar. Local Light Field Fusion: Practical View Synthesis with Prescriptive Sampling Guidelines. *TOG*, 38 (4), 2019. 3

[11] Ben Mildenhall, Pratul P. Srinivasan, Matthew Tancik, Jonathan T. Barron, Ravi Ramamoorthi, and Ren Ng. NeRF: Representing Scenes as Neural Radiance Fields for View Synthesis. In *ECCV*, 2020. 3

[12] Minghan Qin, Wanhua Li, Jiawei Zhou, Haoqian Wang, and Hanspeter Pfister. LangSplat: 3D Language Gaussian Splatting. In *CVPR*, pages 20051–20060, 2024. 2

[13] 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. 1

[14] Qiuhong Shen, Xingyi Yang, and Xinchao Wang. FlashSplat: 2D to 3D Gaussian Splatting Segmentation Solved Optimally. In *ECCV*, pages 456–472. Springer, 2024. 3

[15] Mingqiao Ye, Martin Danelljan, Fisher Yu, and Lei Ke. Gaussian Grouping: Segment and Edit Anything in 3D Scenes. In *ECCV*, 2024. 2, 3

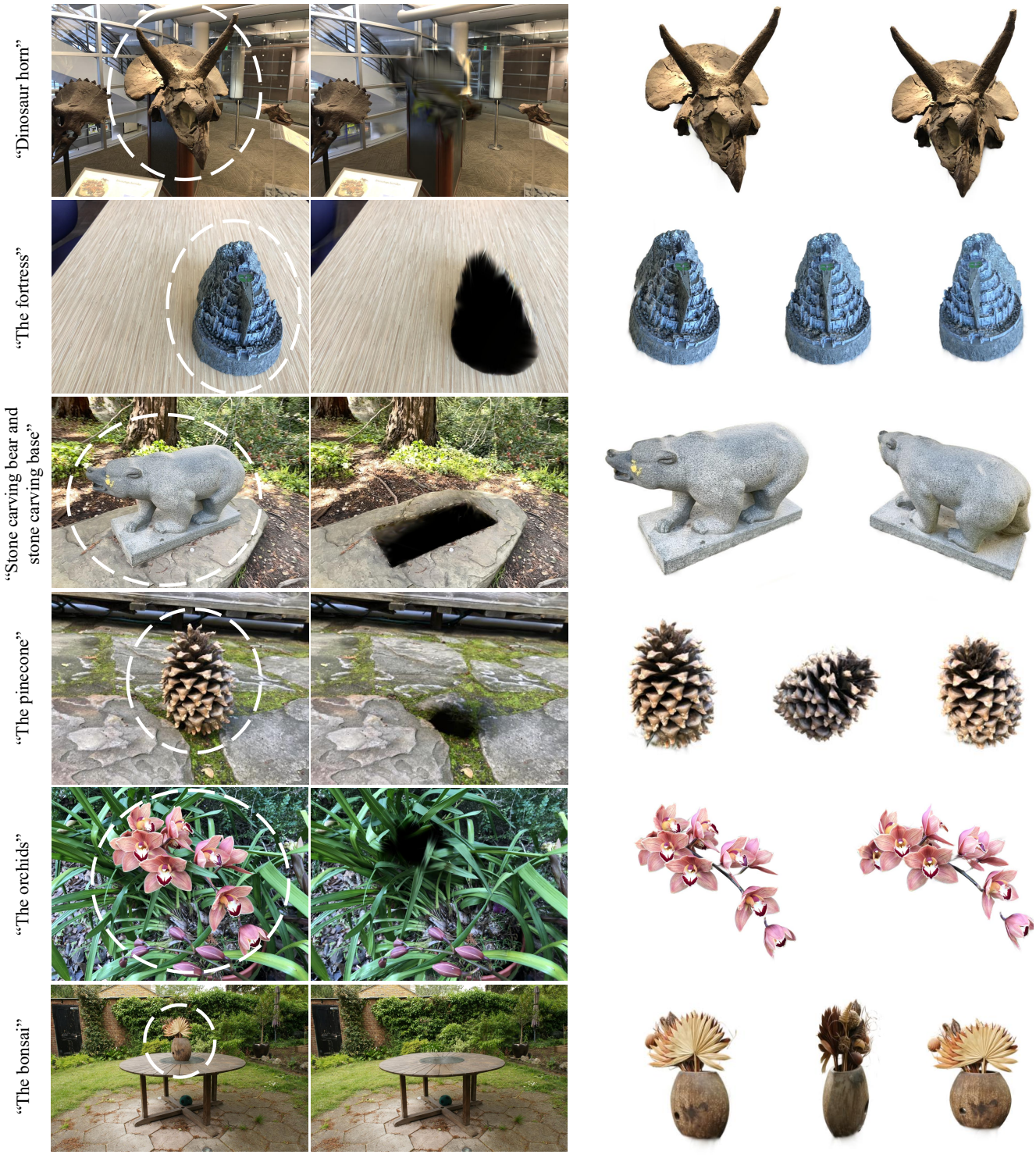

Figure 10. Visualization of 3DGS segmentation. We utilize text prompts to obtain object masks and perform 3D segmentation across multiple scenes, including Horns, Orchids and Fortress from the LLFF dataset, Garden from MIP-360, Bear from the IN2N dataset, and Pinecone from NeRF.