BEA-GS: BEyond RAdiance Supervision in 3DGS for Precise Object Extraction Supplementary Material

Table 1. Sensitivity analysis on the number of points (k) used to compute the voxel density. Extracted metrics (PSNR, IoU, BIoU) across Mip-NeRF 360, LeRF, LLFF, and 3DOVS.

k	Mip-NeRF 360			LeRF			LLFF			3DOVS		
	PSNR	IoU	BIoU	PSNR	IoU	BIoU	PSNR	IoU	BIoU	PSNR	IoU	BIoU
250	29.07	92.0	85.3	24.98	90.4	85.3	24.83	93.0	79.9	26.47	93.3	87.6
500	29.08	92.0	85.6	25.00	90.1	84.9	24.85	93.0	80.2	26.48	93.3	87.5
1000	29.09	92.0	85.7	25.02	90.0	84.5	24.86	93.0	80.7	26.48	93.3	87.5
2000	29.10	92.0	85.8	25.03	89.4	83.6	24.87	93.0	80.7	26.49	93.2	87.3
3000	29.10	92.0	85.8	25.04	88.9	82.9	24.87	92.9	80.7	26.49	93.2	87.2
4000	29.10	92.0	85.8	25.04	88.8	82.6	24.87	92.9	80.6	26.49	93.1	87.2
5000	29.10	91.9	85.8	25.04	88.3	82.0	24.88	92.9	80.5	26.49	93.0	87.0

1. Hyperparameter Sensitivity Analysis

We have performed a sensitivity analysis for k, the number of sampled points to obtain the density of the point cloud, and for Z, the number of points that are sampled from each Gaussian when computing the occupancy loss $\mathcal{L}_{occ}$. Our detailed analysis for k is presented in Tab. 1, and for Z in Tab. 2. The sensitivity of k in our approach is very small, it can be seen that there is nearly no change in rendering quality ($PSNR$). When analyzing the boundary metrics ($IoU, BIoU$), the differences are also minor with small oscillations between datasets. We chose a value that slightly balances these changes. When assessing the sensitivity of Z we see the same behavior, no real difference in rendering quality, and an even smaller oscillation on boundary metrics. The only limitation comes from sampling a number of points bigger or equal than 50 as we run out of memory in the GPU in some scenes. We chose $Z = 20$, but any value that does not overload the VRAM of the GPU is fine as can be observed in Tab. 2. It can be seen that the selection of both k and Z has a very small impact on performance, which shows that our method is robust.

2. Depth Robustness Analysis

To test robustness under degraded geometric supervision, we evaluated performance under noisy depth and surface estimation obtained by retraining the 2DGS backbone without depth and normal regularization losses (Fig. 1). We then applied BEA-GS using these degraded depth maps. Tab 3 reports extracted IoU and BIoU for the original setting (BEA-GS) and the degraded-geometry setting (BEA-Noisy). On datasets with stable reconstructions (MipNeRF360, LLFF), performance remains consistent. On more challenging datasets with weaker capture conditions (LeRF, 3DOVS), we observe a moderate performance drop, indicating that reconstruction quality affects performance, but not to the extent of causing optimization failure.

Table 2. Sensitivity analysis of the number of sampled points Z. Extracted metrics (PSNR, IoU, BIoU) across Mip-NeRF 360, LeRF, LLFF, and 3DOVS. OOM stands for *Out Of Memory*, and thus, the method could not be run.

Z	Mip-NeRF 360			LeRF			LLFF			3DOVS		
	PSNR	IoU	BIoU	PSNR	IoU	BIoU	PSNR	IoU	BIoU	PSNR	EIoU	BIoU
1	29.10	92.0	85.7	25.03	89.4	83.6	24.87	93.0	80.6	26.50	93.2	87.2
2	29.10	92.0	85.7	25.04	89.3	83.5	24.87	93.0	80.6	26.49	93.3	87.3
5	29.10	92.0	85.8	25.04	89.4	83.5	24.87	93.0	80.6	26.49	93.1	87.4
10	29.10	92.0	85.8	25.04	89.4	83.6	24.88	93.0	80.6	26.49	93.3	87.3
20	29.10	92.0	85.8	25.03	89.4	83.6	24.87	93.0	80.7	26.49	93.2	87.3
50	29.10	92.0	85.8	25.03	89.4	83.6	OOM			OOM		

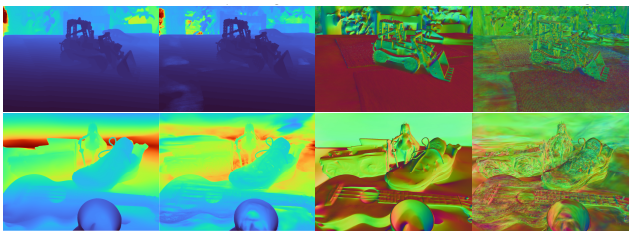

Figure 1. 2DGS Depth (regularized vs. unregularized), 2DGS Normals (regularized vs. unregularized)

Table 3. Depth robustness analysis. Extracted metrics (IoU, BIoU) across Mip-NeRF 360, LeRF, LLFF, and 3DOVS.

	MipNeRF360		LeRF		LLFF		3DOVS	
Method	IoU	BIoU	IoU	BIoU	IoU	BIoU	IoU	BIoU
BEA-GS	92.0	85.8	89.4	83.6	93.0	80.7	93.2	87.3
BEA-Noisy	92.3	86.1	88.4	81.6	92.8	80.7	89.8	83.9

3. SAM2 Reprojection Analysis

We visualize reprojection by coloring each 2D pixel with its class assignment probability (Fig. 2), where brighter col-

ors indicate higher consensus. These probabilities correspond to the per-view reprojection before computing $M' = \text{argmax}(M_\phi)$ (Fig. 3, main paper). In the left example, thin or occluded parts (e.g., eggs, pork) are missed by SAM2 in some views, yielding low-confidence regions, but reprojection recovers coherent boundaries when the object is correctly segmented in at least 50% of frames in which it is visible. Transparent elements (e.g., glass) mainly cause noise due to depth ambiguity. In the bonsai scene (Fig. 2, right), dark low-texture areas near the pot base show consistent SAM2 failures across views, leading to regions that reprojection cannot correct. For the baseline without reprojection (Tab 3, row 4, main paper), BEA-GS remains competitive and reprojection yields the smallest performance change.

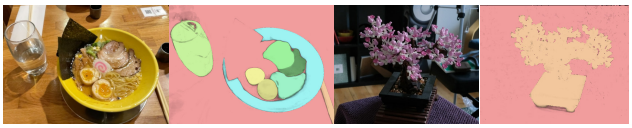

Figure 2. SAM2 probability masks after reprojection

4. Occupancy and Boundary Loss Weights

Although selected empirically, the loss weights can be explained by different gradient scaling behaviors. RGB and boundary losses accumulate gradients over many pixels per Gaussian, yielding larger magnitudes, whereas the occupancy loss is normalized by both Gaussians and sampled points, producing much smaller gradients. Hence, a larger weight ($\lambda_{\text{occ}} = 10$) is needed for balance.

5. Empty Voxels

Our occupancy query checks not only the queried voxel but also its spatial neighborhood. Therefore, even if a voxel is marked as empty due to collisions between multiple semantic classes, sampled points are not penalized as long as valid supporting voxels exist in the local neighborhood, a visualization is shown in Fig. 3. This tries to mitigate undesired erosion effects at object contact regions and provides robustness to artifacts introduced by voxelization.

6. Object Masks Annotation

In order to perform a quantitative evaluation on all datasets (Mip-NeRF 360 [2], LLFF [7], LeRF [5], and 3DOVS [6]) across the two metrics used in our paper, we created the necessary ground-truth segmentation masks. The rendered-view masks were kept exactly as provided by the original datasets as they already segmented the visible regions of the objects and were sufficient for the rendered metrics. The only exception is Mip-NeRF 360, for which no object masks existed; in this case, we generated both the rendered and extracted masks from scratch. Additional masks were required only for the extracted metrics, which need occlusion free silhouettes capturing the full geometry of each object. To construct these, we first extracted the target objects using Trace3D [10] and used the resulting renders as a visual guide in Adobe Photoshop [1] to create the mask. The Trace3D outputs helped us determine how the masks should be filled, and all final mask boundaries were traced and refined manually. LLFF [7] contains no significant occlusions, so the NVOS [9] original masks were used for both rendered and extracted metrics. For LeRF [5], we used the rendered masks provided by Gaussian Grouping [11] and manually created only the additional extracted masks for occluded test views, guided by Trace3D when necessary. The same procedure was followed for 3DOVS [6], reusing the authors' rendered masks and manually generating the extracted ones where needed. For Mip-NeRF 360, both rendered and extracted masks were manually created in Photoshop, again using Trace3D extractions as a reference. Examples of both rendered and extracted masks are shown in Fig. 4.

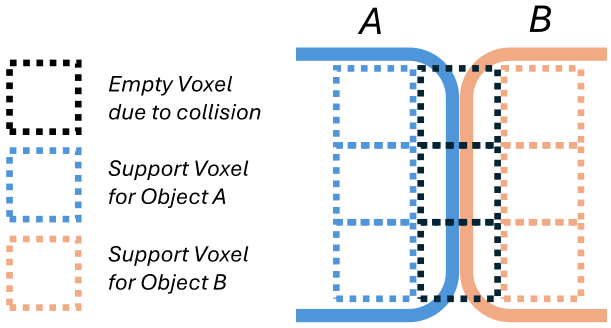

Figure 3. Visualization of edge cases where two objects are in contact. Because voxel validity is determined via neighborhood density checks, erosion effects are mitigated at contact interfaces, ensuring sample points that happens in empty voxels (indicated by the black box) remain valid.

7. Implementation Details

The 2D Boundary Loss is implemented by modifying both the *forward.cu* and *backward.cu* CUDA kernels in the original rasterizer. In contrast, the 3D Boundary Loss is implemented entirely in PyTorch [8]. The 3K training follows the same learning-rate schedule used in the original 2DGS [3] between iterations 27K and 30K. For training, LeRF [5] was used at its original resolution, Mip-NeRF 360 [2] was downsampled by a factor of 4, and both 3DOVS [6] and LLFF [7] were resized to 1.6k pixels width following the standard 3DGS [4] preprocessing procedure. All baselines were optimized using the same input data and the same initial masks for all scenes. Each baseline was optimized using its original training schedule. All methods already included their own mechanism for selecting Gaussians in 3D, so we

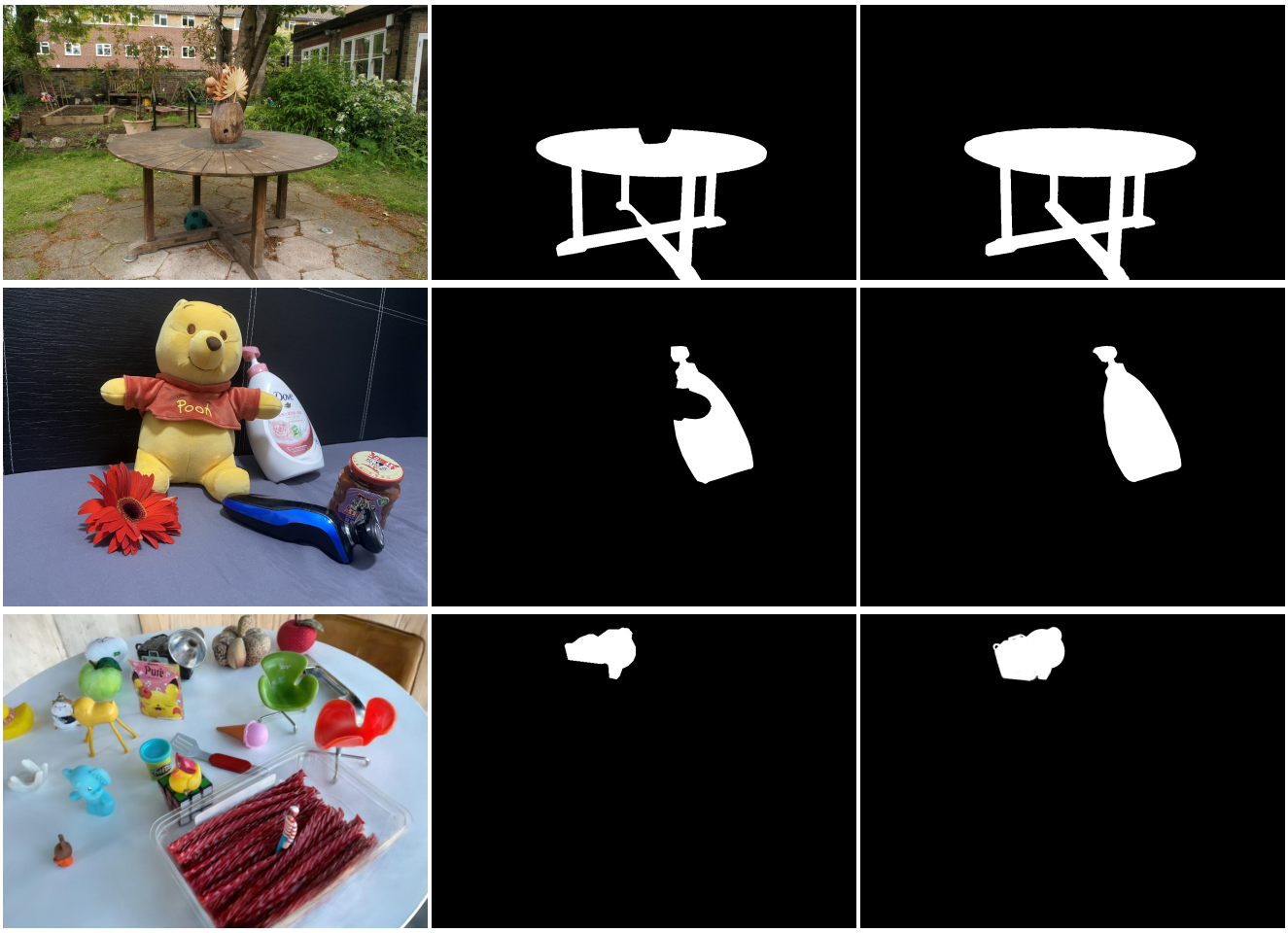

Figure 4. Ground Truth (GT) segmentation masks for both rendered and extracted metrics. *First column:* Original RGB image of the scene. *Middle column:* GT segmentation mask used for rendered metrics, *Last column:* GT segmentation mask used for extracted metrics. The first row shows an example of our GT generated masks for the Mip-NeRF 360 dataset. The second row shows the rendered GT segmentation mask provided by [6], the GT extracted mask was created by us as described in Sec 6. The third row shows the GT rendered segmentation mask provided by [11] for the LeRF dataset, the GT extracted mask was created by us as described in Sec 6.

used their implementations directly without introducing any additional components or modifications specific to our evaluation. We computed all metric means by first averaging the scores over all masks within each scene, and then averaging these scene-level means across the entire dataset.

8. Computational Complexity Discussion

Given the heterogeneous backbones used across the methods we compare to, a direct runtime comparison is not straightforward. We therefore evaluate boundary refinement methods in terms of the additional iterations they require. ObjectGS [13] is the most efficient, performing boundary refinement within the original 30000 training iterations. Trace3D [10] introduces 9000 additional refinement iterations. COB-GS [12], however, trains a separate model for each object instance, causing the number of iterations to scale with both the number of images and the number of classes, making it inefficient for multi-class scenes. For example, in the Figurines scene from LeRF (300 images, 7 classes), COB-GS requires $300 \times 14 \times 7 = 29400$ iterations, compared to 9000 for Trace3D and 3000 for our method, which remains independent of the number of classes.

9. Additional Qualitative Results

Due to space constraints in the main paper, we were unable to include as many qualitative results as desired. In Fig. 5, Fig. 6, and Fig. 7, we therefore present additional objects extracted by our method, showing three rendered viewpoints for each scene. These examples demonstrate that our approach produces high-quality boundary segmentations across all four datasets: Mip-NeRF 360 [2], LLFF [7], LeRF [5], and 3DOVS [6]. For every scene, we remove

the background and visualize the remaining object classes: multiple objects for LeRF and 3DOVS, and a single object for Mip-NeRF 360 and LLFF. The results highlight the method's ability to recover fine-grained details, such as the T-Rex ribcage and the Lego bonsai leaves, which are challenging to infer from 2D segmentations alone. They also illustrate that non-visible Gaussians are no longer an issue during object extraction. Additional qualitative comparisons and results are included in the accompanying video.

References

[1] Adobe Inc. Adobe photoshop. 2

[2] Jonathan T Barron, Ben Mildenhall, Dor Verbin, Pratul P Srinivasan, and Peter Hedman. Mip-nerf 360: Unbounded anti-aliased neural radiance fields. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 5470–5479, 2022. 2, 3

[3] Binbin Huang, Zehao Yu, Anpei Chen, Andreas Geiger, and Shenghua Gao. 2d gaussian splatting for geometrically accurate radiance fields. In *ACM SIGGRAPH 2024 conference papers*, pages 1–11, 2024. 2

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

[5] 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*, pages 19729–19739, 2023. 2, 3

[6] 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. *Advances in Neural Information Processing Systems*, 36:53433–53456, 2023. 2, 3

[7] 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. *ACM Transactions on Graphics (ToG)*, 38(4):1–14, 2019. 2, 3

[8] Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, Alban Desmaison, Andreas Köpf, Edward Yang, Zach DeVito, Martin Raison, Alykhan Tejani, Sasank Chilamkurthy, Benoit Steiner, Lu Fang, Junjie Bai, and Soumith Chintala. Pytorch: An imperative style, high-performance deep learning library, 2019. 2

[9] Zhongzheng Ren, Aseem Agarwala, Bryan Russell, Alexander G Schwing, and Oliver Wang. Neural volumetric object selection. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 6133–6142, 2022. 2

[10] Hongyu Shen, Junfeng Ni, Yixin Chen, Weishuo Li, Mingtao Pei, and Siyuan Huang. Trace3d: Consistent segmentation lifting via gaussian instance tracing. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 6656–6666, 2025. 2, 3

[11] Mingqiao Ye, Martin Danelljan, Fisher Yu, and Lei Ke. Gaussian grouping: Segment and edit anything in 3d scenes. In *European conference on computer vision*, pages 162–179. Springer, 2024. 2, 3

[12] 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 *Proceedings of the Computer Vision and Pattern Recognition Conference (CVPR)*, pages 19335–19344, 2025. 3

[13] 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 *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 8350–8360, 2025. 3

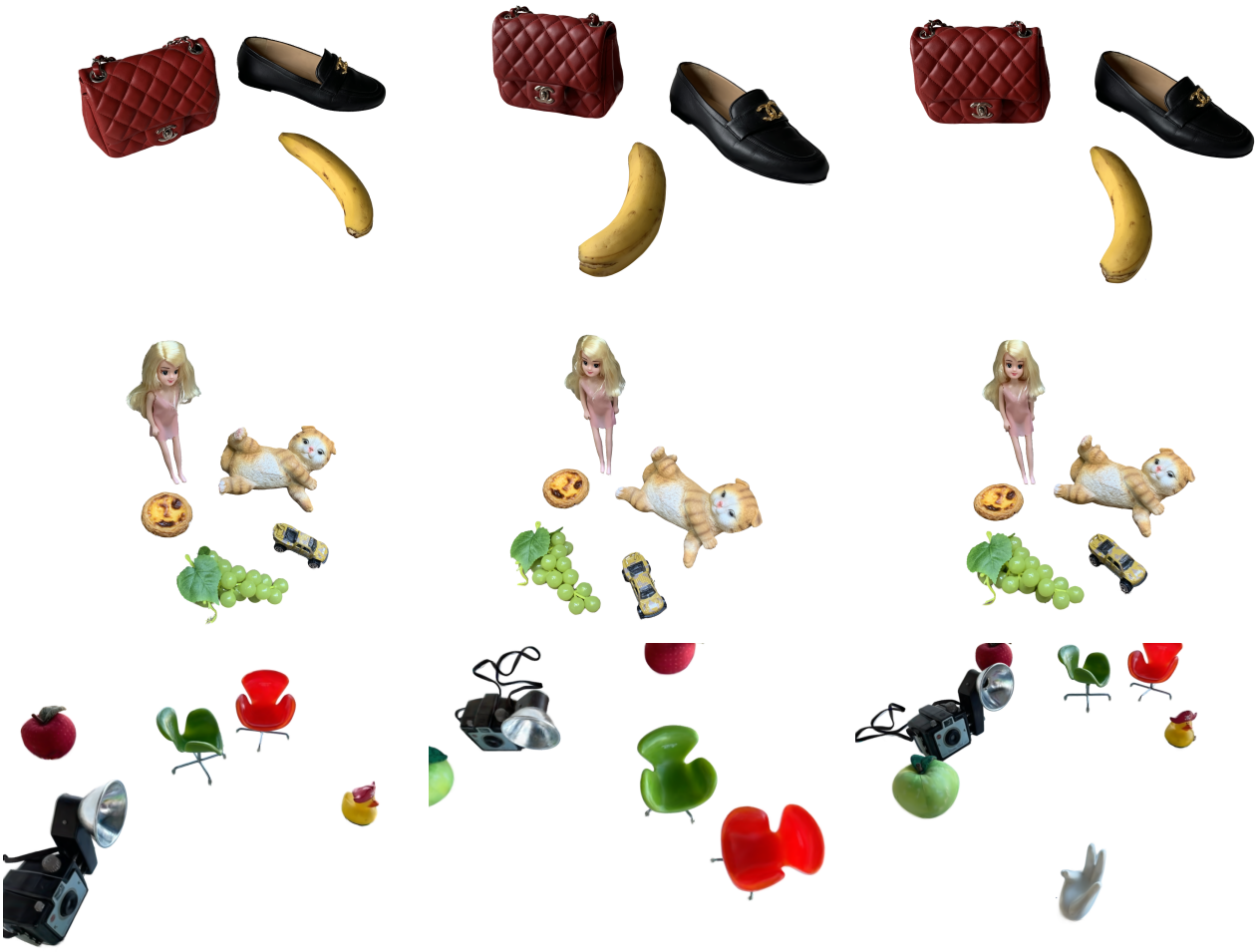

Figure 5. Additional qualitative results obtained using our proposed approach

Figure 6. Additional qualitative results obtained using our proposed approach

Figure 7. Additional qualitative results obtained using our proposed approach