Trace3D: Consistent Segmentation Lifting via Gaussian Instance Tracing

Supplementary Material

A. Failure Cases and Limitations

Failure Cases We visualize typical failure cases in Fig. S.1. In Case 1, the artifact Gaussians are small and located far from the painting's surface. When traced from the forward view, it belongs to the painting, whereas from the lateral view, it belongs to the wall. In both views, this Gaussian only belongs to one object, albeit different ones. As a result, it is not considered ambiguous and we cannot eliminate them despite being clear artifacts. Case 2 and 3 share similar issues. Our Gaussian Instance Tracing (GIT) is consistent with the rendering process and terminates when the accumulated opacity reaches 1. Therefore, if a Gaussian belongs to an object and is partially covered by a highly opaque surface, it cannot be flagged as ambiguous. These limitations degrade performance on fine-grained details, affecting hierarchical segmentation and object extraction.

Figure S.1. Failure cases on Replica.

Limitations We discuss two limitations that are valuable to address as future work.

First, although our motivation stems from leveraging the inherent consistency of Gaussians as a more explicit representation—ideally assigning one Gaussian per object or part—the approach still relies on a neural implicit framework with alpha blending for rendering. Consequently, there is a gap between true 3D surface geometry and the rendering-based association of Gaussians. As the level of granularity increases, especially for fine-grained details or texture-like patterns, the premise of consistent masks and the assumption about ambiguous Gaussians weaken. This also can be seen from Fig. 1 and Fig. 7 in the main paper, where hierarchical segmentation and object extraction exhibit more artifacts when the detail is refined to parts of a single body (*e.g.*, Gundam) or object (*e.g.*, camera). While our method is capable of robust segmentation and object extract on common objects under diverse scenarios, these findings underscore a recurring dilemma: while implicit representations such as Neural Radiance Fields (NeRF) [7], Gaussian Splatting [2, 3], and DMTet [9] are relatively easy to optimize, fully explicit representations are often better suited for physics-based rendering and efficient simulation.

Second, although the GIT operation is comparable in speed to forward rendering, merging inconsistent maps remains a relatively time-consuming step, as detailed in Appendix B.2. Future research could investigate more efficient data structures and algorithms to further mitigate training overhead. Nevertheless, our Nonetheless, our GIT-guided density control works as efficient as the original density control in original 3D Gaussian Splatting (3DGS) [3].

B. Method Details

B.1. Scalability of GIT

The weight matrix $\mathbf{W} \in \mathbb{R}^{N \times T \times L}$, where N is the number of Gaussians, T the maximum number of patches in any single view, and L the number of views, is conceptually defined but not explicitly used during implementation. In practice, we accelerate GIT by first computing a temporary $N \times T$ matrix sequentially for each view, which is then reduced to an N-dim vector storing the patch ID with the highest probability. The vectors for all L views are combined into a $N \times L$ matrix for patch merging and GS refinement. Thus, our method only maintains a temporary $N \times T$ matrix and a cumulative $N \times L$ matrix during training, allowing us to handle scenes with a large number of objects efficiently.

B.2. Scalability of patch merging

This step involves computing patch similarity by tracing all relevant Gaussians and checking whether they belong to the same patch based on majority votes across all views. The theoretical upper bound on complexity is $O(N^2 \cdot T^2)$. While in implementation, we avoid computing similarities for all primitive-level patch pairs within a view based on SAM's hierarchical information: only patches that fall within the same region of coarser-level masks are considered for merging. Furthermore, patch similarity is computed only over co-visible views shared by the two Gaussian sets, rather than across all views. Combined with our GIT acceleration, they ensure efficiency in large-scale, multi-object scenes.

C. Method Comparison

Previous work [1, 11] also proposes an overlay mask solution to alleviate inconsistencies in multi-view masks generated by SAM [5]. Here, we provide a detailed discussion comparing our method with these approaches.

EgoLifter EgoLifter [1] discards information about overlapping regions and simply overlays all masks in image

Table S.1. The selected id lists used for 3D object extraction experiment in Replica.

Scene	ID list
office0	3,4,7,9,11,12,14,15,16,17,19,21,22,23,29,30,32,34,35,36,37,40,44,48,49,57,58,61,66
office1	3,8,9,11,12,13,14,17,23,24,29,30,31,32,34,35,37,43,45
office2	0,2,3,4,6,8,9,12,13,14,17,23,27,34,38,39,46,49,51,54,57,58,59,63,65,68,69,70,72,73,74,75,77,78 80,84,85,86,90,92,93
office3	1,2,8,11,12,15,18,21,22,25,29,32,33,42,51,54,55,56,60,61,70,82,85,86,88,86,97,101,102,103,110,111
office4	3,4,5,6,9,13,16,18,20,23,31,34,47,48,49,51,52,56,60,61,62,65,69,70,71
room0	1,2,3,4,6,7,8,11,13,15,18,19,20,21,22,24,30,32,34,35,36,39,40,41,43,45,47,49,50,51,54,55,58, 61,63,64,68,69,70,71,72,73,74,75,78,79,83,85,86,87,90,92
room1	3,4,6,7,8,9,11,12,13,15,17,18,19,21,22,23,24,27,30,32,33,35,37,39,40,43,45,46,48,50,51,52,53,54
room2	2,4,5,10,14,15,17,18,19,20,22,24,26,27,28,29,31,32,34,36,38,39,40,42,44,46,47,48,49,52,54,55,56, 57,58,59,61

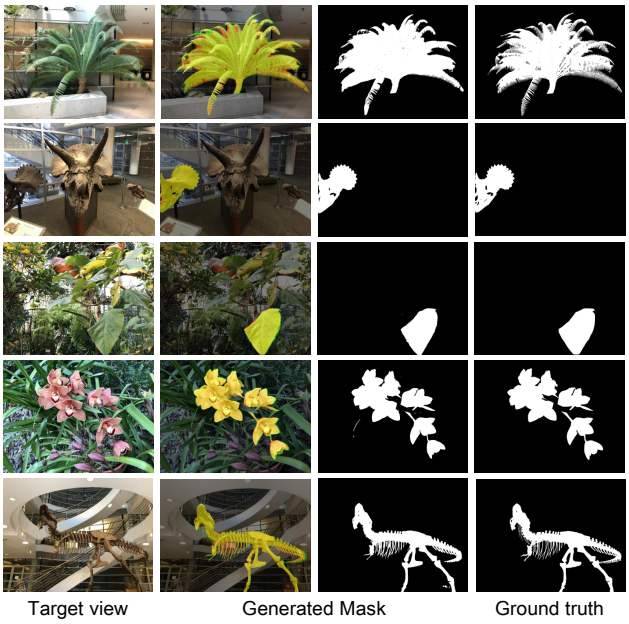

Figure S.2. Remaining visualization results on NVOS.

space to obtain a one-hot segmentation for each pixel. However, due to the randomness of the topmost mask across different views, EgoLifter's solution fails to resolve inconsistencies in the overlapping areas.

Omniseg3D Similarly to our approach, OmniSeg3D [11] also divides the 2D image into disjoint patches by overlapping the SAM masks. it models the hierarchical structure within these patches by measuring correlations, characterized by the number of masks that contain the corresponding masks. During contrastive lifting, it enforces this hierarchy through an explicit ordering regularization. While the concept is promising, inconsistent SAM predictions can produce ambiguous features across different hierarchical levels, especially in fine-grained cases. This issue is evident in the qualitative comparisons in the main paper.

GarField GarField [4] optimizes a scale-conditioned affinity field in an attempt to alleviate inconsistencies in multi-view masks. However, its scale conditioning is overly sensitive and operates as a black-box, making it challenging to use during the inference phase.

Gaga While both our method and Gaga [6] leverage the mask-Gaussian relationship, Gaga focuses on 3D lifting with mask ID association, whereas our method addresses inconsistent SAM masks and refines Gaussians, benefiting general 3D segmentation lifting methods. Gaga traces via Gaussian center projection, which struggles with occlusion, while our reverse rasterization offers more precise, view-consistent alignment. Moreover, Gaga uses a sequential 3D memory bank, where assignments are fixed once added, limiting effective cross-view aggregation. In contrast, our majority voting across all views enables more robust and consistent segmentation.

Ours We obtain a single instance map by overlapping all masks and treating each overlapping region as a disjoint instance patch. We then perform consistent masking and GIT-guided density control on these primitive-level patches, providing clear guidance during contrastive lifting. This ensures that features are distinctly grouped or separated, leading to sharper boundaries, clearer feature maps, and flexible segmentation or object extraction across various levels of granularity in our experiments.

D. Experiments: 3D Object Extraction

D.1. Evaluation Details

We render the extracted 3D object into novel views to generate its corresponding RGB image. The regions where the RGB values exceed zero are utilized as a mask to compute

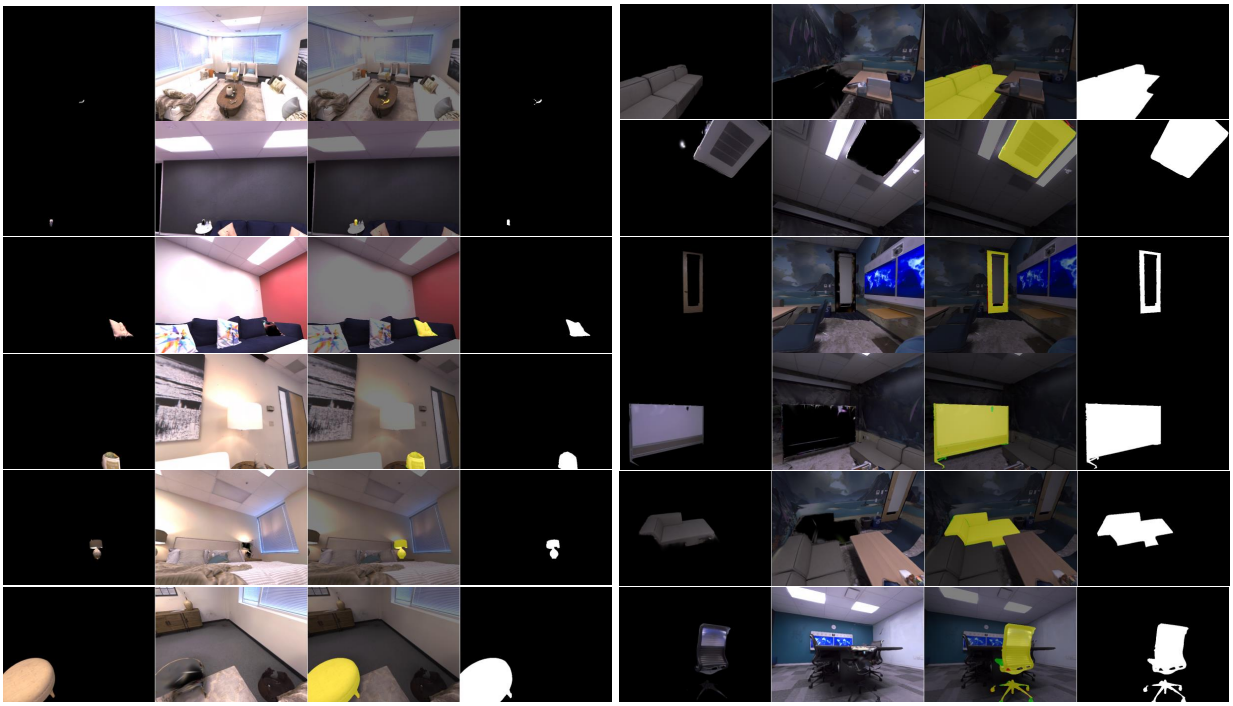

Figure S.3. More visualization results on Replica.

Table S.2. Quantitative results of 3D object extraction on Replica across all scenes.

Method	office0	office1	office2	office3	office4	room0	room1	room2	*avg.* mIoU	*avg.* PSNR
Gaussain Grouping [10]	23.7	45.9	25.5	30.6	30.2	22.5	38.5	20.1	29.6	13.4
FlashSplat [8]	47.5	45.9	39.6	36.9	27.5	40.6	39.8	36.6	39.3	16.9
Egolifter [1]	67.4	59.6	48.9	54.8	59.4	50.7	53.7	50.1	55.6	20.1
Gaga [6]	45.1	47.8	37.8	37.2	40.4	39.3	36.9	44.5	41.1	17.6
Ours	**80.7**	**76.0**	**66.1**	**69.8**	**71.7**	**67.0**	**72.0**	**73.4**	**72.1**	**22.6**

Table S.3. The detailed PSNR of 3D object extraction on Replica across all scenes.

Method	office0	office1	office2	office3	office4	room0	room1	room2	*avg.* PSNR
Gaussain Grouping [10]	16.0	22.8	10.5	11.2	12.3	10.3	12.7	11.7	13.4
FlashSplat [8]	21.2	24.5	14.4	14.0	14.9	14.8	15.2	16.2	16.9
Egolifter [1]	26.5	29.1	16.3	16.0	19.5	17.0	17.4	19.1	20.1
Gaga [6]	22.4	27.0	13.5	13.9	16.7	14.5	14.7	18.0	17.6
Ours	**28.1**	**29.9**	**18.9**	**20.0**	**22.0**	**19.4**	**20.3**	**22.3**	**22.6**

the IoU with the ground-truth mask. For PSNR calculation, we restrict the computation to a specific region defined by the bounding box of the ground-truth mask, expanded outward by 10 pixels. This ensures that the evaluation focuses on the relevant object regions while reducing the influence of background areas.

D.2. Testing Split

We select the majority of instances from the Replica dataset, excluding floors, ceilings, excessively large walls, and low-quality objects. The full list is provided in Tab. S.1. To more accurately evaluate the quality of the extracted 3D objects, we filter out test views where the objects are occluded.

E. More Experiment Results

We provide detailed results on Replica in Tab. S.2 Tab. S.3. Additionally, we present further qualitative results on Replica in Fig. S.3 and on the remaining NVOS scenes in Fig. S.2.

References

[1] Qiao Gu, Zhaoyang Lv, Duncan Frost, Simon Green, Julian Straub, and Chris Sweeney. Egolifter: Open-world 3d segmentation for egocentric perception. In *European Conference on Computer Vision (ECCV)*, 2025. 1, 3

[2] 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*, 2024. 1

[3] Bernhard Kerbl, Georgios Kopanas, Thomas Leimkühler, and George Drettakis. 3d gaussian splatting for real-time radiance field rendering. *ACM Transactions on Graphics (TOG)*, 2023. 1

[4] Chung Min Kim, Mingxuan Wu, Justin Kerr, Ken Goldberg, Matthew Tancik, and Angjoo Kanazawa. Garfield: Group anything with radiance fields. In *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2024. 2

[5] Alexander Kirillov, Eric Mintun, Nikhila Ravi, Hanzi Mao, Chloe Rolland, Laura Gustafson, Tete Xiao, Spencer Whitehead, Alexander C Berg, Wan-Yen Lo, et al. Segment anything. In *International Conference on Computer Vision (ICCV)*, 2023. 1

[6] Weijie Lyu, Xueting Li, Abhijit Kundu, Yi-Hsuan Tsai, and Ming-Hsuan Yang. Gaga: Group any gaussians via 3d-aware memory bank. *arXiv preprint arXiv:2404.07977*, 2024. 2, 3

[7] 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. *Communications of the ACM*, 2021. 1

[8] Qiuhong Shen, Xingyi Yang, and Xinchao Wang. Flashsplat: 2d to 3d gaussian splatting segmentation solved optimally. In *European Conference on Computer Vision (ECCV)*, 2025. 3

[9] Tianchang Shen, Jun Gao, Kangxue Yin, Ming-Yu Liu, and Sanja Fidler. Deep marching tetrahedra: a hybrid representation for high-resolution 3d shape synthesis. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2021. 1

[10] Mingqiao Ye, Martin Danelljan, Fisher Yu, and Lei Ke. Gaussian grouping: Segment and edit anything in 3d scenes. In *European Conference on Computer Vision (ECCV)*, 2024. 3

[11] Haiyang Ying, Yixuan Yin, Jinzhi Zhang, Fan Wang, Tao Yu, Ruqi Huang, and Lu Fang. Omniseg3d: Omniversal 3d segmentation via hierarchical contrastive learning. In *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2024. 1, 2