Rethinking End-to-End 2D to 3D Scene Segmentation in Gaussian Splatting

Runsong Zhu[1] Shi Qiu[1] Zhengzhe Liu[2,3] Ka-Hei Hui[1] Qianyi Wu[4]
Pheng-Ann Heng[1] Chi-Wing Fu[1]
[1]The Chinese University of Hong Kong [2] Lingnan University
[3] Carnegie Mellon University [4] Monash University

Supplementary

In the supplementary material, we provide more technical details, experimental results, and downstream applications of our proposed method:

- Sec. A: implementation details.
- Sec. B: quantitative & qualitative comparisons and ablation studies.
- Sec. C: downstream applications, including object Gaussian extraction, object removal, and object mesh reconstruction.
- Sec. D: limitation and future work.

A. Implementation details

We implement our method based on the 3D Gaussian Splatting (3D-GS) representation [4]. Our method comprises three main modules: the basic 3D Gaussian splatting backbone, the Gaussian-level features, and a global object-level codebook. For the basic 3D Gaussian splatting backbone, we follow the official 3D Gaussian Splatting implementation in [4]. For Gaussian-level features, we set the dimension of the Gaussian-level feature to 16 for a fair comparison with previous baselines (e.g., Gaussian Grouping [10] and OmniSeg3D-GS [11]). By using the splatting technique, the rendered feature map results in a shape of $16 \times H \times W$, where 16 is the feature dimension and (H, W) represents the resolution of (Height, Width). For the object-level codebook, we set the default value of L as 256, representing the maximum number of objects in the scene, and $d = 16$ applies the same feature dimension as in Gaussian-level features.

During training, we jointly train the basic 3D Gaussian Splatting backbone, the Gaussian-level features, and the object-level codebook. Specifically, we utilize the same image rendering loss with the same learning rate and density control as outlined in [4] to optimize the original 3D-GS backbone. We employ the contrastive loss term (Eq. 1 in the main paper) to optimize the Gaussian-level feature and use the Adam optimizer with a learning rate of 0.0025. Moreover, we use the association constraints (Eq. 8 in the main paper) to optimize the object-level codebook and employ the Adam optimizer with a learning rate of 0.0005. In practice, we block the gradient derived from the association constraints from propagating to the Gaussian-level features. This gradient-blocking design ensures that the Gaussian-level features are exclusively optimized through the contrastive loss. We jointly train all parameters by directly adding all the loss terms—image rendering loss, contrastive loss, and association constraints—with equal weight (*i.e.*, 1) for 30,000 iterations on each dataset covered in this work, using a single NVIDIA RTX 3090.

B. Experimental results

B.1. More results

3D visual results. We provide 3D Gaussian center point-level visual comparisons on the LERF-masked dataset. Following the protocol in Gaussian Grouping, we obtain the selected instance IDs and highlight them in 3D. The results in Fig. 1 further show that our new framework is able to produce more accurate 3D segmentation.

Visualization for ablation study. The visual results in Fig. 2 demonstrate that both the explicit codebook formulation and the tailored codebook training strategies progressively enhance the effectiveness of our end-to-end lifting framework.

Complete comparisons on Messy Room dataset. We provide the results of Gaussian Grouping [10] in Tab. 1 for complete comparisons on the Messry Room dataset [2]. The results demonstrate that our method achieves clearly improved performance than the existing segmentation method based on 3D Gaussian representation.

Time efficiency. We provide a detailed comparison of the time efficiency in Tab. 1. Overall, our method significantly enhances performance and requires a similar amount of total time (approximately 1 hour) as existing efficient baselines. Thanks to our lightweight object-level codebook

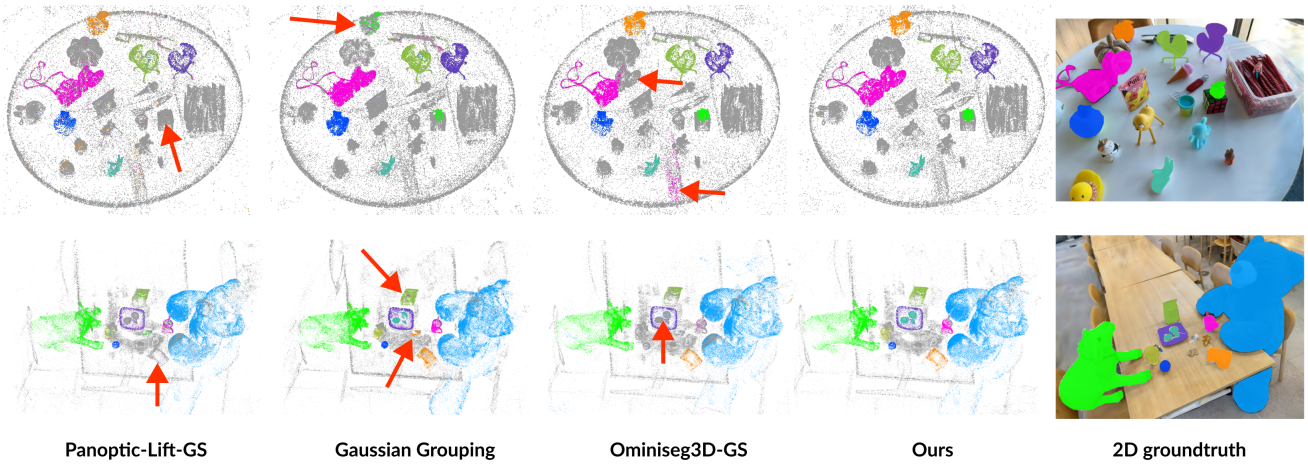


Figure 1. 3D results on LERF-masked dataset.

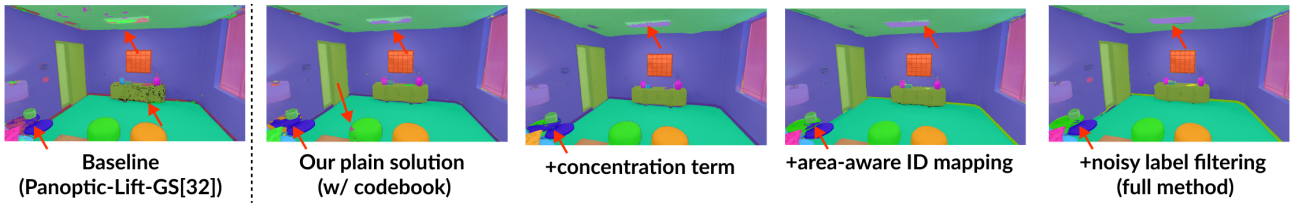


Figure 2. Visual results for ablation study on Replica.

Type	Method	Mean PQ^{scene}(%)	Pre-processing	Training	Post-processing	Total time
Pre-processing	Gaussian Grouping [10]	48.9	7 min	44 min	0	51 min
End-to-end	Panoptic-Lifting-GS [6]	58.6	0	56 min ($\approx 1h$)	0	56 min
Post-processing	OmniSeg3D-GS [11]	66.0	0	57 min ($\approx 1h$)	24 min	81 min
Post-processing	OmniSeg3D-GS (best hyper-parameter for all scenes) [11]	65.4	0	57 min ($\approx 1h$)	24 min	81 min
End-to-end	Ours	69.0	0	62 min ($\approx 1h$)	0	62 min

Table 1. Detailed time efficiency analysis on the Messy Rooms dataset [2]. Note that, we test the time (including pre-processing, training, and post-processing) for all methods using a single NVIDIA RTX 3090 GPU. Following [2], PQ^{scene} metric is reported for comparisons.

design, the joint learning of Gaussian-level features and the object-level codebook does not increase the time budget much. The pre-processing method, Gaussian Grouping [10], achieves the best training speed but suffers from inferior performance. For the post-processing method, the results in Tab. 1 show that hyper-parameter tuning in clustering is time-consuming, requiring an additional 24 minutes per scene on the Messy Room dataset[2], which further limits its applicability.

Detailed comparisons with post-processing methods. For the post-processing lifting method, *i.e.*, OminiSeg3D-GS [11], we report the performance under the optimal best-found hyper-parameter (*i.e.*, minimal cluster size) for HDBSCAN clustering algorithm [5], following the same strategy used in Contrastive Lift [2]. Specifically, we utilize the training views to search for the best hyper-parameter for each scene, setting the search range from 10 to 200, as suggested in "Tuning Clustering Hyperparameter" [2]. In contrast, our new end-to-end lifting pipeline enjoys the following two strengths: effectiveness and efficiency. **1). Effectiveness.** We plot the results of OminiSeg3D-GS [11] under different parameter values (minimal cluster size) in Fig. 5, showing that the post-processing is sensitive to the hyper-parameter. Moreover, while the exhaustive search can improve performance, it is still behind our method, which achieves consistent results without the need for hyperparameter tuning. As shown in Fig. 3, we notice that it is challenging to achieve a good balance between avoiding artifacts and identifying small objects by tuning hyper-parameters in the clustering algorithm. This challenge inherently limits the upper potential of the post-processing lifting method. **2). Efficiency.** Tab. 1 indicates that the hyper-parameter tuning in post-processing is time-consuming, leading to an additional 30% of the total training time compared to our pipeline.

Qualitative comparisons. We provide more qualitative comparisons in Fig. 6, Fig. 7 and Fig. 8. These visual results further demonstrate that our method delivers more

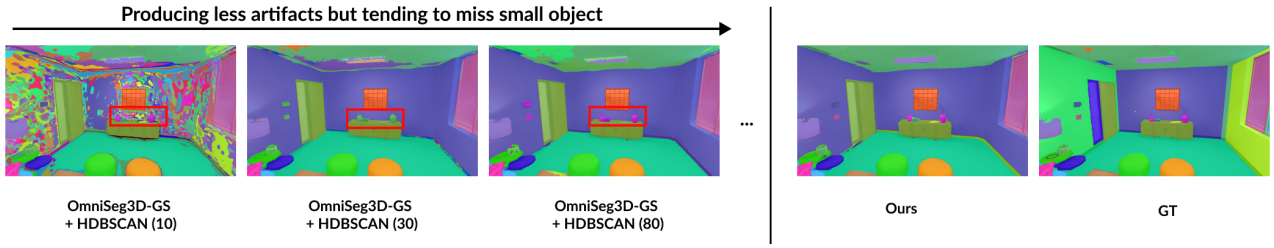


Figure 3. Qualitative comparison of post-processing method (*i.e.*, OminiSeg3D-GS [11]) and our method on replica dataset [10]. We plot the results of OmniSeg3D-GS [11] under different hyper-parameters (i.e., minimal cluster size), which demonstrate the difficulty in achieving a good balance between avoiding artifacts and identifying small objects by tuning the hyperparameters.

accurate and consistent segmentation across various views, while also minimizing artifacts.

Method	LERF-masked		Replica		Training
	mIoU	mBIoU	mIoU	F-score	time
Contrastive Lift	75.0	72.4	38.2	34.0	≥ 20 h
Ours	**80.9**	**77.1**	**41.6**	**43.9**	≈ 1 h

Table 2. More quantitative comparisons with Contrastive Lift.

Figure 4. Visual comparison with Contrastive Lift on Replica.

More comparisons with Contrastive Lift [2]. We provide more comparisons with the NeRF-based post-processing baseline Contrastive Lift [2]. We perform experiments for Contrastive Lift on the LERF-masked and our customized Replica datasets as no public results are available for Contrastive Lift. The results in Tab. 2 and additional visualizations in Fig. 4 show that our method clearly outperforms Contrastive Lift in terms of segmentation results and training time, again confirming its effectiveness.

B.2. Ablation Study

We further conduct the following ablation studies on the Replica dataset [10] for a better analysis of our method.

ID matching strategy	mIoU(%)	F-score(%)
Mapping in Panoptic Lifting [6]	30.3	30.4
Proposed area-aware ID mapping	31.7	33.5

Table 3. Effectiveness analysis of the proposed area-aware ID mapping method. We compare the segmentation results of pseudo-labels generated by our area-aware ID mapping and the approach proposed in Panoptic Lifting [6].

Area-aware ID mapping. To further verify the effectiveness of our area-aware ID mapping, we present additional quantitative comparisons between the pseudo-labels generated by our area-aware ID mapping method and the method proposed in Panoptic Lifting [6]. The results shown in Tab. 3 verify that the pseudo-labels generated by our area-aware ID mapping are more accurate and consistent.

τ	0.75	0.80 (default)	0.85
mIoU(%)	40.0	41.6	40.4
F-score(%)	43.0	43.9	43.7

Table 4. Quantitative comparisons of using different thresholds τ values in the noisy label filtering module.

Method	mIoU(%)	F-score (%)
Full model	41.6	43.9
W/o gradient-blocking design	39.7	39.8

Table 5. Ablation study on the effectiveness of our gradient-blocking design.

Model	PSNR↑	SSIM↑	LPIPS↓
3D Gaussian Splatting [4]	28.69	0.870	0.182
Gaussian Grouping [10]	28.43	0.863	0.189
Ours	28.53	0.869	0.180

Table 6. Quantitative comparisons of rendering quality on Mip-NeRF360 dataset [1]. Our method utilizes the same 3D-GS backbone as in Gaussian Grouping [10], which achieves almost the same performance for novel-view synthesis. We provide the results for completeness.

Sensitivity to different per-defined values in noisy label filtering. We investigate the impact of the predefined threshold used in the noisy label filtering module. In our main experiments, we set a predefined threshold of $\tau = 0.8$ to filter noisy segmentations in the noisy label filtering module. To investigate the impact of this threshold, we conduct additional experiments using two different values ($\tau = 0.75$ and 0.85). As shown in Tab. 4, the results remain rather stable despite moderate changes in the threshold τ.

Gradient-blocking. The results in Tab. 5 demonstrate that our gradient-blocking strategy design improves opti-

mization stability.

Rendering quality. We utilize the same 3D-GS backbone as Gaussian Grouping [10], and provide the rendering quality comparisons on the Mip-NeRF 360 dataset [1] in Tab. 6. As expected, our method achieves almost the same performance for novel-view RGB image synthesis as Gaussian Grouping and 3D-GS.

C. Application results

We conduct three downstream applications to demonstrate the effectiveness of our method.

Object Gaussian extraction. Based on the explicit 3D GS representation, we can perform the 3D object Gaussian extraction utilizing the 3D segmentation results. Specifically, we directly identify the corresponding Gaussians for the selected object IDs. Then, we utilize the extracted object Gaussian to perform novel-view synthesis for visual comparison with Gaussian Grouping [10] in Fig. 9 (a). The results illustrate that our method achieves accurate object-level Gaussian asset extraction with fewer noisy Gaussians.

Object removal. We further test the application of object removal. In practice, we directly remove the corresponding Gaussians belonging to the selected object ID and render the image with the remaining Gaussians. As shown in Fig. 9 (b), our method can generate clearer results compared with Gaussian grouping, due to the fact that more accurate 3D segmentation results are exploited.

Object mesh reconstruction. We showcase that our method can further facilitate object mesh reconstruction. Concretely, we incorporate the multi-view consistent segmentation maps generated by our method and Gaussian Grouping [10] into the 2D-GS pipeline [3], and extract the 3D mesh belonging to the selected object. We provide a visual comparison in Fig. 10. The results show that our segmentation results can lead to better object meshes with more accurate boundaries and fewer artifacts, further demonstrating the effectiveness of our method.

D. Limitation and future work

Our method focuses on static 3D scene segmentation for 3D understanding and cannot handle scenes with dynamic objects. To further enable 4D scene understanding, one possible direction is to incorporate segmentation information into the 4D-GS methods [8, 9]. We leave this for future work.

References

[1] 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. 3, 4

[2] Yash Bhalgat, Iro Laina, João F Henriques, Andrew Zisserman, and Andrea Vedaldi. Contrastive Lift: 3D object instance segmentation by slow-fast contrastive fusion. *arXiv preprint arXiv:2306.04633*, 2023. 1, 2, 3, 5

[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. 4, 10

[4] Bernhard Kerbl, Georgios Kopanas, Thomas Leimkühler, and George Drettakis. 3D gaussian splatting for real-time radiance field rendering. *ACM Transactions on Graphics*, 42 (4), 2023. 1, 3

[5] Leland McInnes, John Healy, and Steve Astels. hdbscan: Hierarchical density based clustering. *J. Open Source Softw.*, 2(11):205, 2017. 2

[6] Yawar Siddiqui, Lorenzo Porzi, Samuel Rota Bulò, Norman Müller, Matthias Nießner, Angela Dai, and Peter Kontschieder. Panoptic lifting for 3D scene understanding with neural fields. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 9043–9052, 2023. 2, 3

[7] Julian Straub, Thomas Whelan, Lingni Ma, Yufan Chen, Erik Wijmans, Simon Green, Jakob J Engel, Raul Mur-Artal, Carl Ren, Shobhit Verma, et al. The replica dataset: A digital replica of indoor spaces. *arXiv preprint arXiv:1906.05797*, 2019. 5, 7, 8

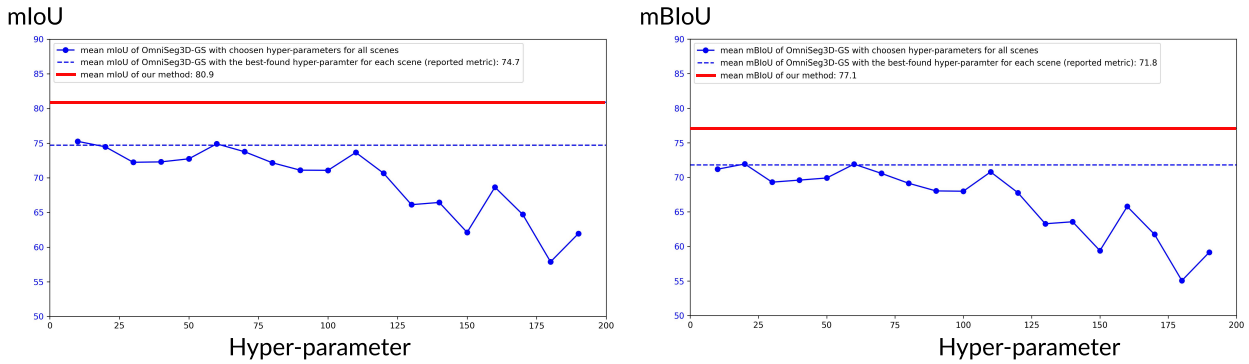
[8] Guanjun Wu, Taoran Yi, Jiemin Fang, Lingxi Xie, Xiaopeng Zhang, Wei Wei, Wenyu Liu, Qi Tian, and Xinggang Wang. 4d gaussian splatting for real-time dynamic scene rendering. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 20310–20320, 2024. 4

[9] Zeyu Yang, Hongye Yang, Zijie Pan, and Li Zhang. Real-time photorealistic dynamic scene representation and rendering with 4d gaussian splatting. *arXiv preprint arXiv:2310.10642*, 2023. 4

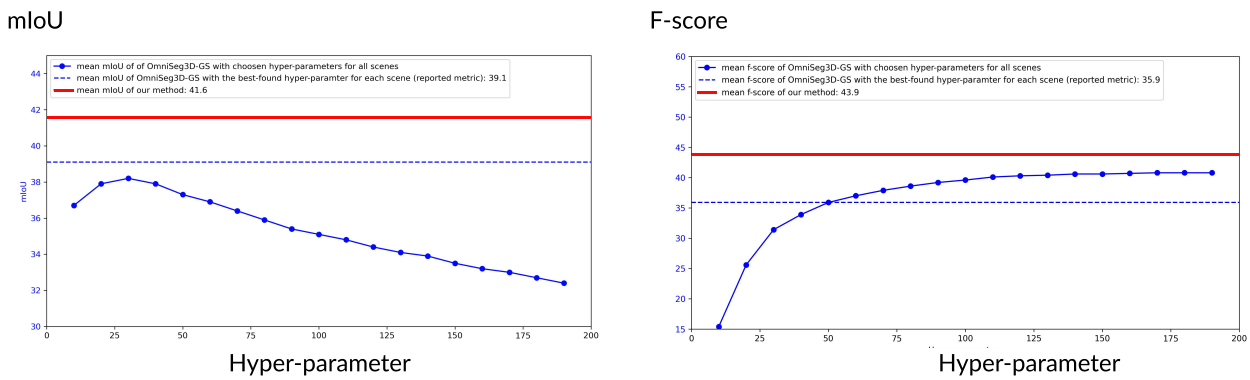
[10] Mingqiao Ye, Martin Danelljan, Fisher Yu, and Lei Ke. Gaussian grouping: Segment and edit anything in 3D scenes. *arXiv preprint arXiv:2312.00732*, 2023. 1, 2, 3, 4, 5, 6, 9, 10

[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 *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 20612–20622, 2024. 1, 2, 3

(a) LERF-Masked dataset



(b) Replica dataset



(c) Messy Rooms dataset

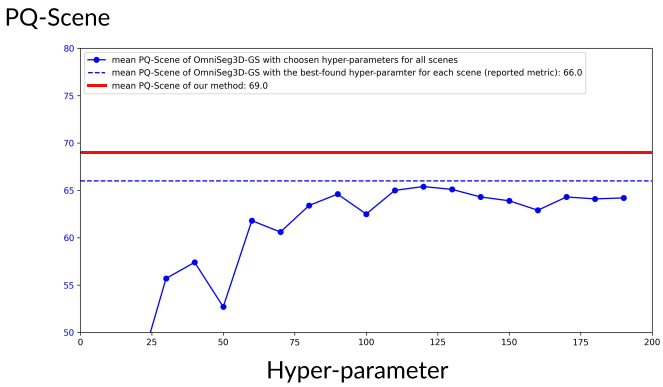


Figure 5. The detailed comparison between our method and OmniSeg3D-GS methods on the LERF-Mask dataset [10], Replica dataset [7] and the Messy Rooms dataset [2] dataset. For LERF and Replica datasets, we utilize the mIoU metric to search the best hyper-parameter for each scene. For the Messy Rooms dataset, we utilize the PQ-Scne to select the best hyper-parameter for each scene. Note that our method is denoted by the red color, and OmniSeg3D-GS is denoted by the blue color.

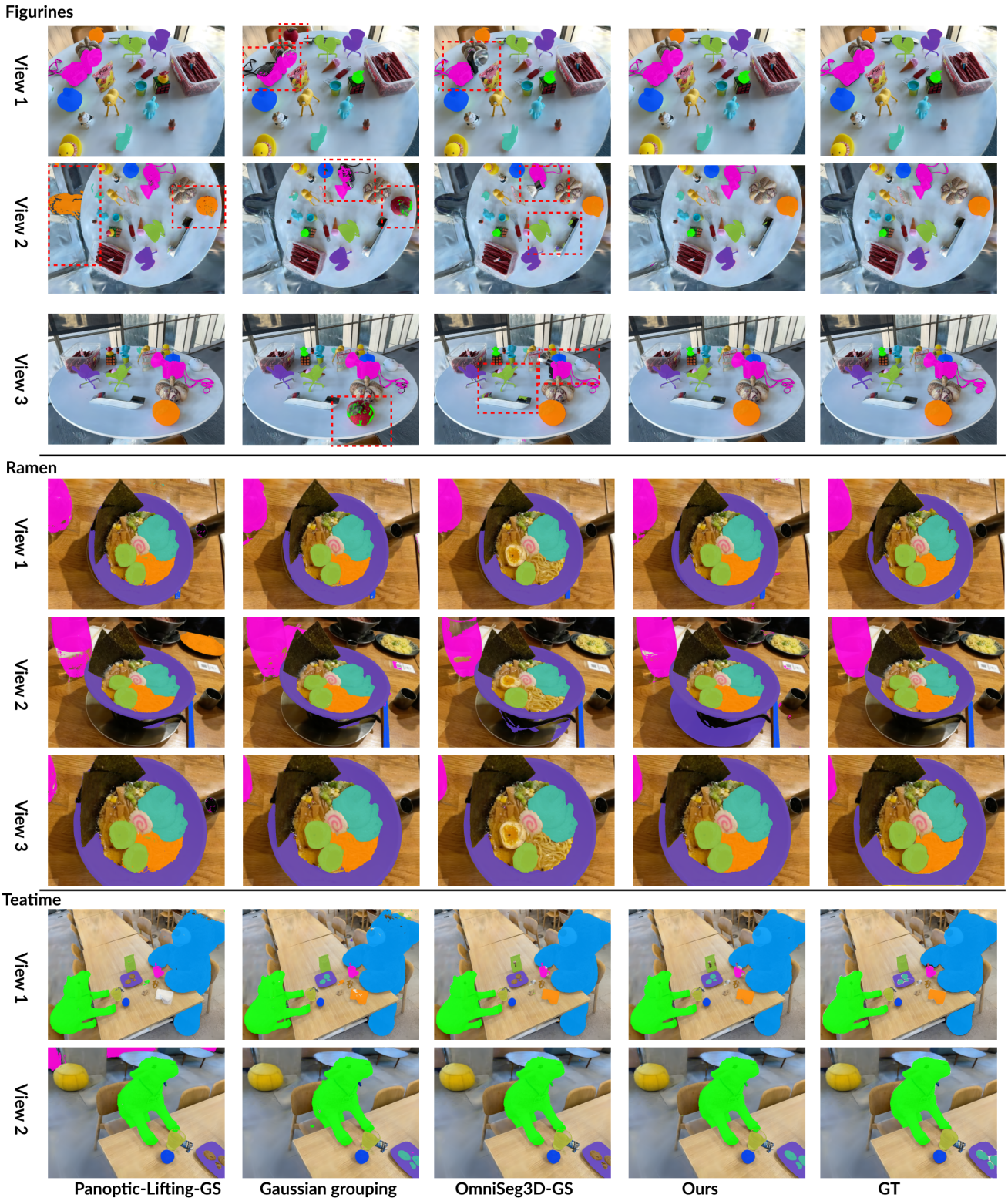


Figure 6. Visual comparisons between our method and previous methods on the LERF-Masked dataset [10].

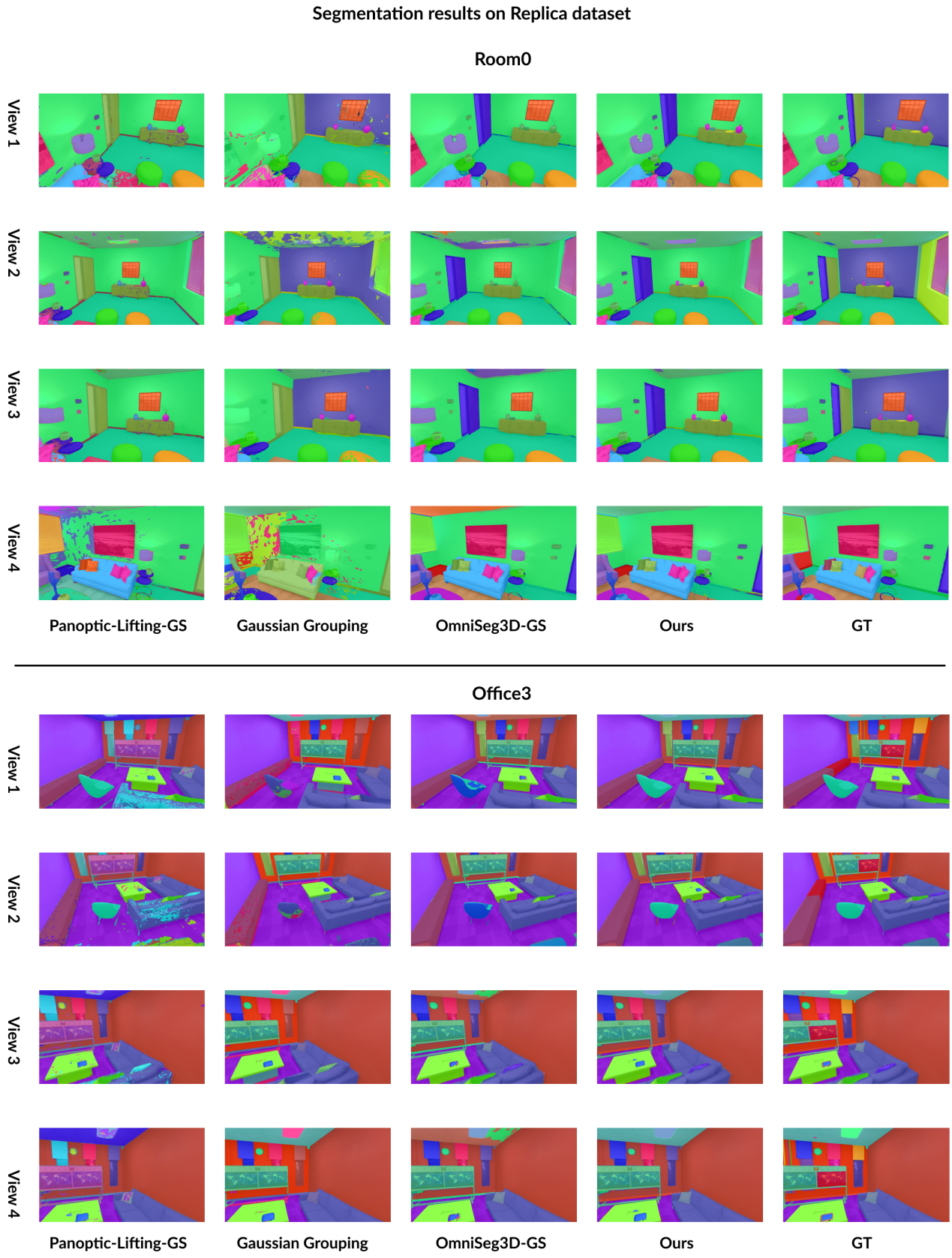


Figure 7. Visual comparisons between our method and previous methods on the Replica dataset [7].

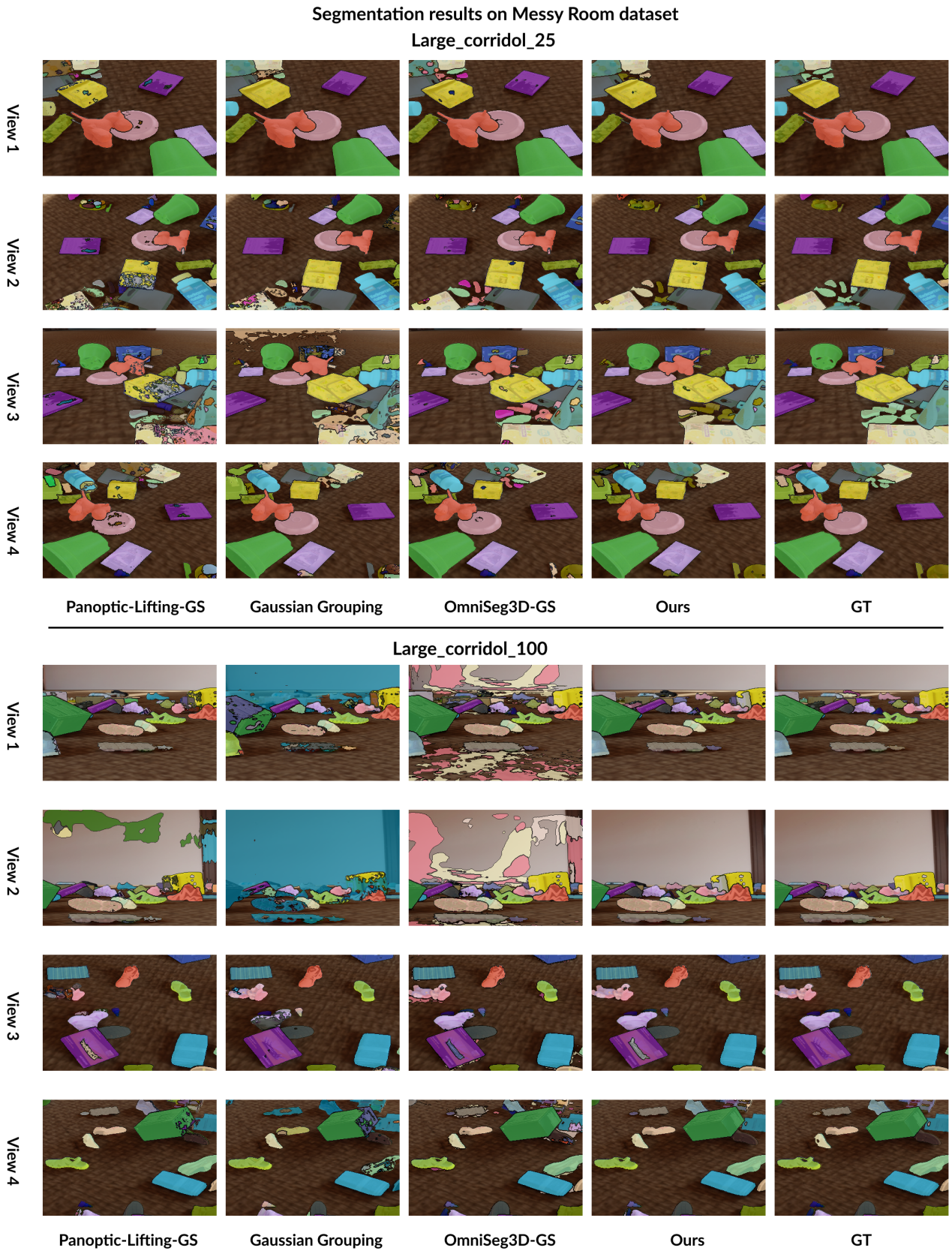


Figure 8. Visual comparisons between our method and previous methods on the Messy Rooms dataset [7].

Figure 9. Qualitative comparison of application results on LERF-Masked dataset [10]. As shown in (a), we provide a visual comparison of object Gaussian assert extraction result with Gaussian grouping [10]. Thanks to the accurate 3D segmentation results, we could extract more accurate Gaussians for the select object ensuring less noisy rendering results. As shown in (b), we provide comparisons for object removal. We remove the Gaussians belonging to the "yellow toy" object and utilize the left Gaussian to perform the novel view synthesis. Notably, to better illustrate the 3D segmentation quality, we do not use post-processing [10] for our method and baseline. The results show that our method could generate more clean images than the results generate by Gaussian Grouping [10]. We use the officially released checkpoint of Gaussian Grouping [10] to conduct the experiments.

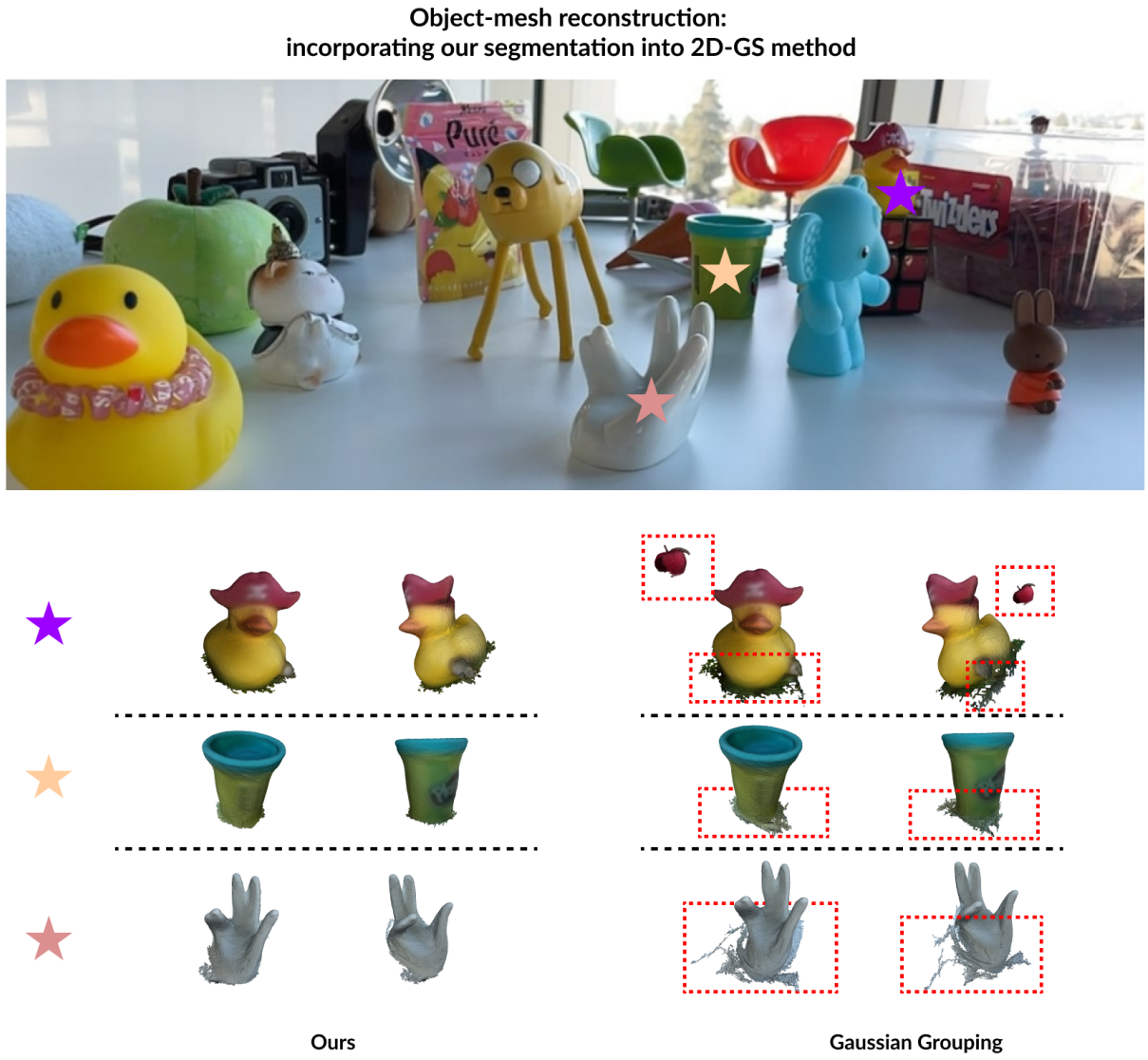


Figure 10. Visualizations of the object mesh reconstruction application results on the LERF-Masked dataset [10]. We demonstrate that our consistent segmentation can be incorporated into the surface reconstruction method (i.e., 2D-GS [3]) to facilitate object-level mesh results. The results show that using our segmentation can lead to object meshes with more accurate boundaries and fewer artifacts. We use the officially released checkpoint of Gaussian Grouping [10] to conduct the experiments.