InstaScene: Towards Complete 3D Instance Decomposition and Reconstruction from Cluttered Scenes

Supplementary Material

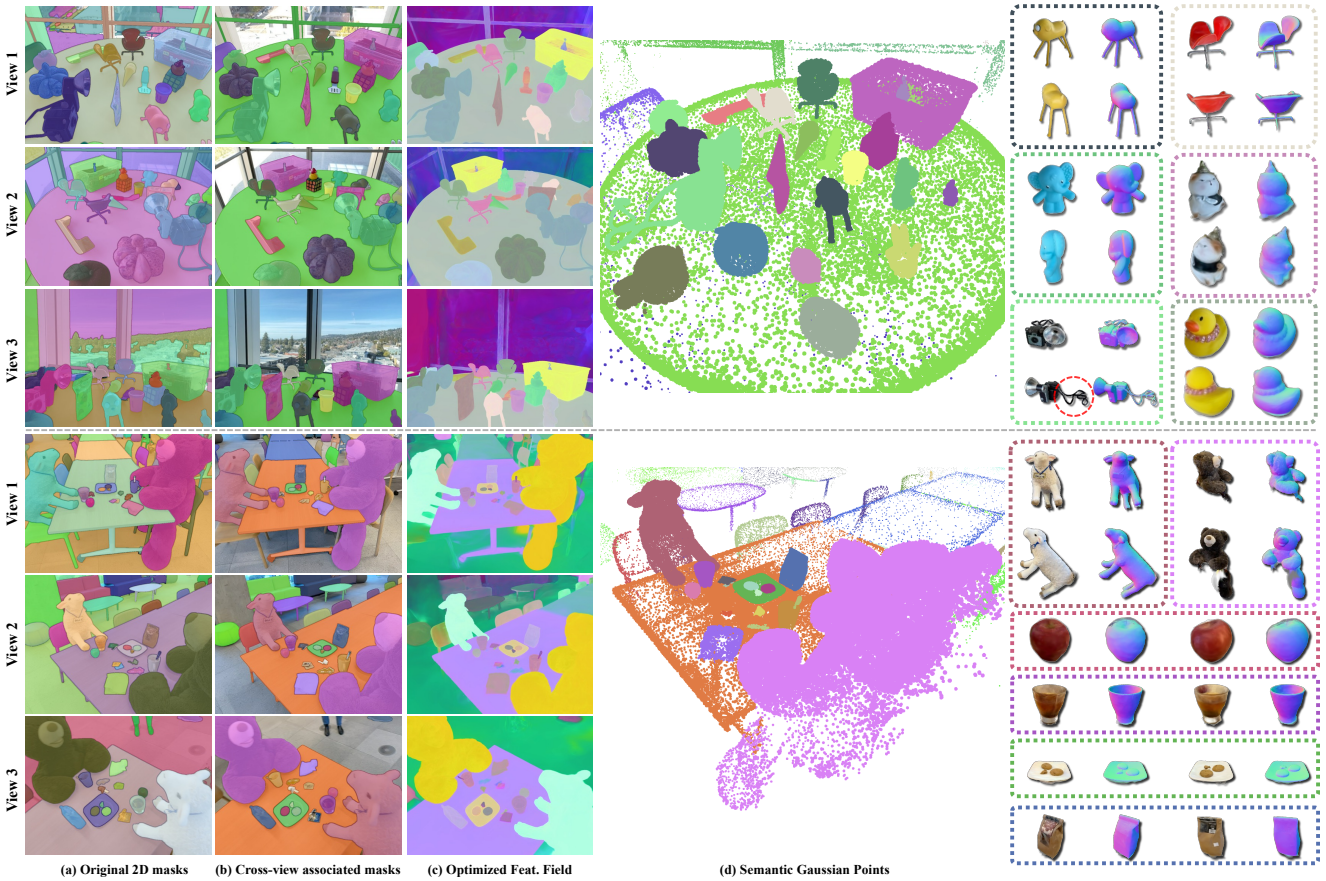


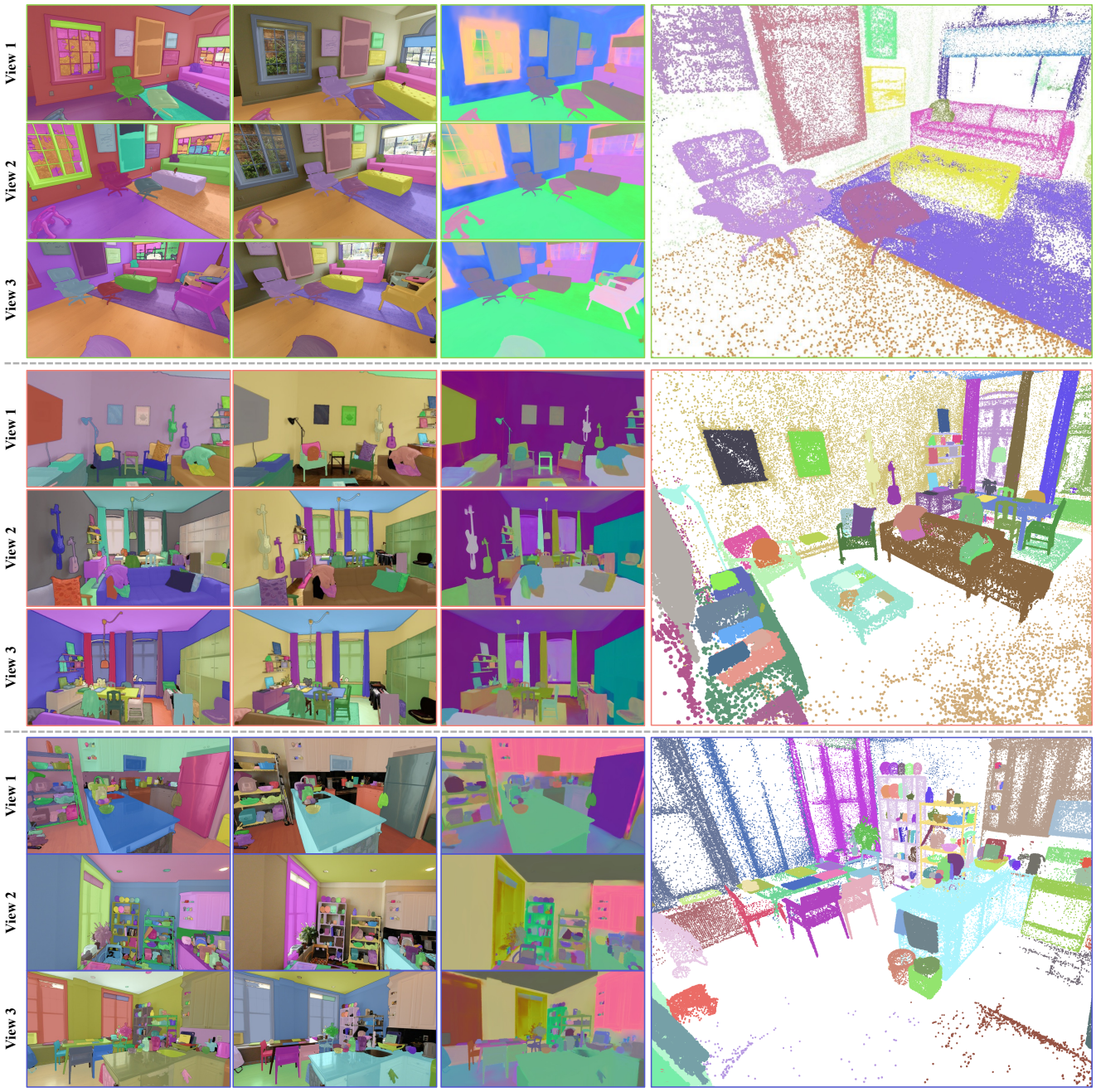
Figure 1. **Novel view rendering for segmented instances.** We present segmentation results on the LERF-Mask Dataset [7], including the original segmentation masks, cross-view associated masks, and the final optimized feature field. Instances with the same label are marked using the same color. Additionally, we extract representative instances from each scene, demonstrating that our method can even capture fine-grained details such as camera straps.

In this supplementary material, we provide more details of our InstaScene framework, including: 1) Implementation details of our fine-grained scene decomposition (Sec. 1); 2) Additional results on various datasets (Sec. 2, 3, 4, 5, 6); 3) Implementation details of our in-situ generation (Sec. 7); 4) Extended quantitative comparison and user study of the in-situ generation (Sec. 8, 9). 5) Failure cases arising from the limitation of our method (Sec. 10). Additionally, we provide a supplemented video to summarize our method and provide more intuitive visualizations and the demonstration of scene interaction.

1. Implementation Details of Decomposition

We follow the default training settings of 2DGS [5] to obtain pre-trained Gaussian models from the given posed RGB frames of the scene. We use EntitySeg with the CropFormer-Hornet-L backbone [11] to obtain 2D instance segmentation. For filtering under-segmented masks, we consider if the spatial tracker $P_{i,j}$ of mask $m_{i,j}$ intersects with multiple spatial trackers $\{P_k\}$ from the same frame I_k, and the highest overlap rate, defined as $\tau_{overlap} = \frac{\max_l |P_{i,j} \cap P_{k,l}|}{\sum_l |P_{i,j} \cap P_{k,l}|}$ is less than 0.8 of the total intersection area, and this situation occurs in 30% of the frames where $P_{i,j}$ is visible, then the mask $m_{i,j}$ is marked as under-segmented.

We employ the Adam optimizer with a learning rate of 2.5×10^{-2} to train Gaussian's feature. For contrastive learning, we randomly sample 32×1024 pixels on a single view while simultaneously sampling 64×1024 points from the global instance priors $\mathcal{M}_i^{3d}$ that are visible within the current view. For the multi-view supervision, we randomly sample 64×1024 pixels from the current frame and its adjacent views $\{I_j \mid j \in [i-k, i+k]\}$ for every 5 iter-



(a) Original 2D masks (b) Cross-view associated masks (c) Optimized Feat. Field (d) Semantic Gaussian Points

Figure 2. **Additional visualization for decomposition results.** We present the instance segmentation results on the ZipNeRF dataset [1]. We accurately establish cross-view mask associations and subsequently optimize the distinguishable feature field. The final semantic Gaussian points obtained are also illustrated as described above.

ations, where $k = 2$. For the hyperparameters in Eq. 7 of the main paper, we set $\lambda_1 = 1 \times 10^{-6}$, $\lambda_2 = 1 \times 10^{-6}$, $\lambda_3 = 2.5 \times 10^{-6}$. The pre-training process which establishes the cross-view masks with traced Gaussian clustering takes approximately 5 minutes. We randomly initialize the feature embeddings of each Gaussian point $\{\mathbf{f}_i^{3d} \in \mathbb{R}^D\}$ and train the feature field for 10,000 iterations, with the entire process averaging 20 minutes on an NVIDIA Tesla A100-40GB. With the trained GS feature field, our method supports real-time interactive segmentation at 1K resolution with 100+FPS on an RTX 4070-12GB GPU, please see the supp. video for further details.

2. Additional Results for Scene Decomposition

We present additional results on the LERF-Mask Dataset [7] and the ZipNeRF Dataset [1] as shown in Fig. 1, 2. We achieve cross-view mask association based on the traced Gaussian clustering and obtain distinguishable feature fields. Note that not every predicted segmentation mask is retained in the cross-view association as we only

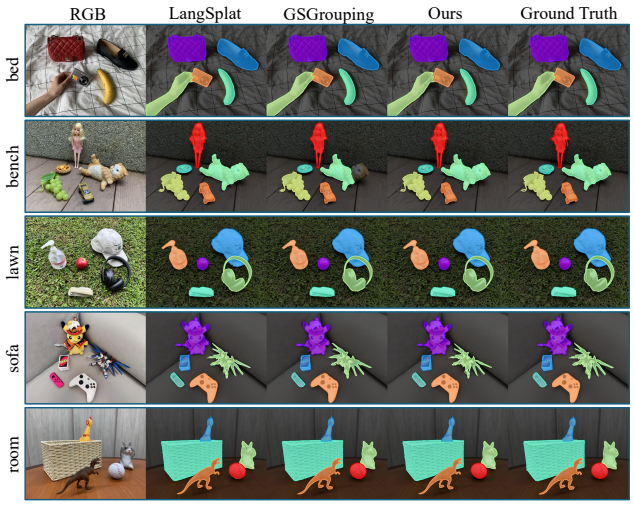


Figure 3. **Additional Comparison on 3D-OVS Dataset.** The 3D-OVS Dataset fails to effectively highlight the differences between different methods due to the simplicity of the scenes.

Method	Bench	Bed	Room	Sofa	Lawn	Average
Langsplat	95.2	97.1	94.3	95.1	95.9	95.5
GSGrouping	89.7	**97.5**	94.5	94.3	94.7	94.1
Ours	**95.3**	96.9	**96.5**	**95.4**	**96.2**	**96.0**

Table 1. **Quantitative comparison on 3D-OVS Dataset.** All methods exhibit favorable performance on the 3D-OVS dataset.

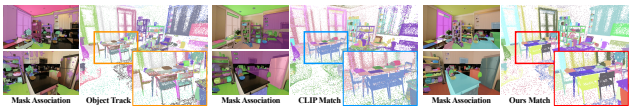


Figure 4. **Comparison with other mask association methods.** Object tracking and CLIP-based matching methods both underperform in cluttered scenes with duplicate objects.

retain masks within each cluster that exhibit a relatively large overlap area with other masks, which helps to filter out unreliable 2D segmentation. We also extract the corresponding Gaussian points of several representative instances from each scene and perform novel view rendering as shown in the last column of Fig. 1. As illustrated in the Figurine scene in Fig. 1, our method successfully preserves fine-grained details, such as the camera strap at the last row marked in red, thereby further demonstrating the high precision of our scene decomposition approach.

3. Comparisons on the 3D-OVS Dataset

Following Langsplat [12], we conduct additional comparisons on the 3D-OVS Dataset [8], which comprises scenes composed of simple objects manually arranged, without repetitive objects or cluttered environments. As the results shown in Fig. 3 and Tab. 1, both our method and the baseline approaches achieve favorable performance. In contrast, as illustrated in Fig. 6 of the main paper, we further conduct experiments on the more complex indoor scenes from the ZipNeRF Dataset, further demonstrating the robustness of our approach.

Figure 5. **Visualization of Edge Cases.** The integration of Gaussian tracker with spatial contrastive learning effectively addresses challenging edge cases involving repetitive or textureless objects, while also mitigating under-segmentation issues present in the 2D segmentation prior.

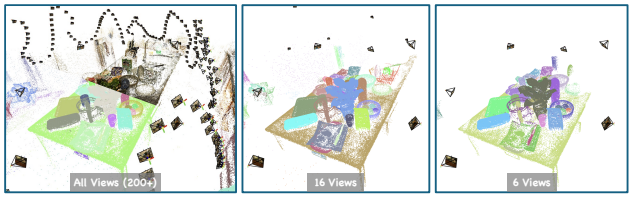


Figure 6. **The 3D segmentation under sparse views input.** Our method is capable of performing accurate 3D scene segmentation even under extremely sparse input views.

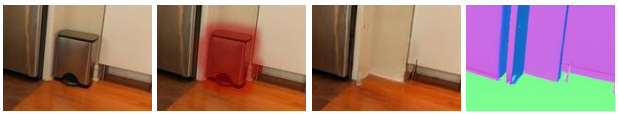


Figure 7. **Visualization of Background Inpainting.**

4. Comparison with Other Mask Association

Our spatial tracker establishes cross-view and global 3D association by tracing spatial relationships among Gaussians, which also filters out 2D under-segmentation. We compare other commonly used cross-view mask association strategies, such as object tracking (e.g., SAM2) and CLIP-based matching. As shown in Fig. 4, both methods underperform in cluttered scenes with duplicate objects.

5. Visualization of More Edge Cases

We present additional results addressing real-world challenges such as repetitive textureless objects and severe occlusions to further demonstrate the robustness of our segmentation method as shown in Fig. 5. Our method is actually the first to explicitly address them by: **1)** Using a strong instance-level 2D segmentation model which inher-

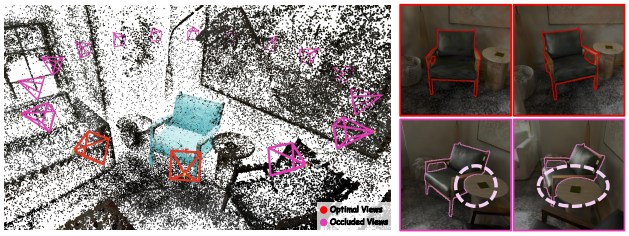


Figure 8. **Optimal viewpoints selection.** The viewpoints marked in red undergo minimal occlusion by the scene and are selected as ideal viewpoints. The viewpoints marked in pink are significantly occluded and we classify them as unseen views that need to be supplemented by the generative model.

ently distinguishes repetitive or textureless objects; **2)** Clustering multi-view consistent 2D segmentation masks and filter under-segmentation (see view3 semantic masks shown in Fig. 5) to ensure reliable supervision; **3)** Learning distinctive features via spatial contrastive learning with the enhanced 2D semantic labels, no feature distillation from DINO & CLIP is required, thus can handle repetitive objects that are frustrating for pre-trained VLMs.

As shown in Fig. 6, our method leverages spatial correlations, enabling robust 3D segmentation even with a very limited number of input views (We first perform Gaussian Splatting reconstruction with dense input views, then evaluate with varying numbers of 2D segmentation maps as input.).

6. Background Inpainting

Although our work primarily focuses on completing unobserved foreground regions, Since background inpainting (object removal) has been widely studied, followed by existing methods (e.g., GSGrouping [17], GSEditor [2], VRGS [6]), given rendered views, we first extract object masks with the learned feature field and feed them into PowerPaint [18] for 2D object removal. The inpainting results are subsequently used to refine the neighboring Gaussians for background inpainting as shown in Fig. 7.

7. Implementation Details of In-situ Generation

Occlusion-aware Optimal View Selection. We present an intuitive visualization of the optimal viewpoint selection for each instance used as input for in-situ generation in Fig. 8. We refer to the target view setup in [10] and select 16 viewpoints centered around the segmented object, from an elevation of 30 degrees, with the azimuth linearly spaced across 360 degrees. Additionally, we adopt the standard intrinsic settings used for rendering [3] as training data [9, 10]. We consider the viewpoint with the least occlusion by the scene as the optimal input. Specifically, given a viewpoint π_i, we render its depth based on the Gaussian points of both

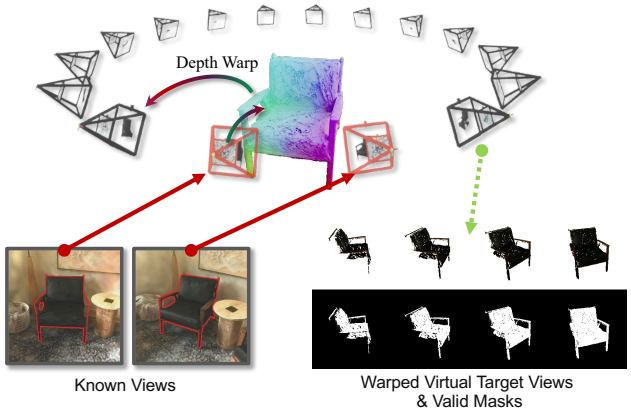


Figure 9. **Complement target views with known features.** We employ geometric projection to map the known observations onto the target view, further constraining the regions with known information and enhancing consistency across generated target views.

the segmented instance and the scene. If 85% of the difference between the instance depth and the scene depth is less than 0.05, we classify the viewpoint as not occluded by the scene and include it in the set of optimal views. The remaining viewpoints are considered unseen and require the generative model to supplement.

Joint Optimization. After obtaining the viewpoints supplemented by the 3D generative prior, we perform joint optimization with the source observations. For the joint optimization, we maintain the same learning rate for each Gaussian parameter as [5] and train each instance for 2,000 steps, which takes approximately 30s on an NVIDIA Tesla A100-40GB.

Complement Known Features via Geometry Cues. As depicted in Fig. 9, we add time-dependent noise to the latent features of the input views, and leverage the known geometry (i.e., rendered depths of input views) to project the known latent features onto the visible regions of the target views, while initializing the invisible regions with random noise. This approach enforces multi-view consistency in the visible regions throughout the diffusion process while simultaneously guiding the denoising of more plausible results in the invisible regions, and also enhances the utilization of available reconstructed information without altering the structure of the generative model.

Alignment with Scene. To align the results generated by the baselines [4, 15, 16] with the original scene, as the relative camera poses are known when generating multiple views [4, 14] in InstantMesh [16] and MVDFusion [4], for MVDFusion, we project the generated RGB-D images into the scene coordinate system according to the relative poses, converting them into pointcloud, and scale it based on the radii of instance. For InstantMesh, we normalize the generated mesh and project it into the scene using the transformation relationship between the actual scene and its coordinate system. Since the coordinate system of SpaRP [15] is un-

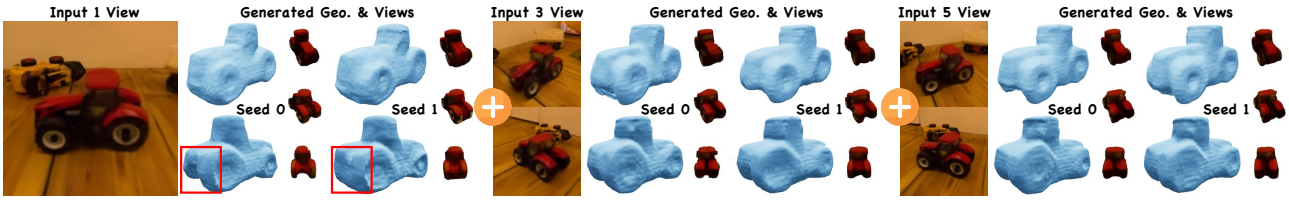


Figure 10. **Ambiguity in completion.** While varying seeds produce different outcomes, the results converge as input views increase.

Methods	InstantMesh [16]	SpaRP [15]	Ours
Alignment ↑	1.936	1.251	**2.813**
Faithful Comp. ↑	2.019	1.398	**2.583**

Table 2. **User Study.** Our method outperforms in both alignment and faithful completion evaluation.

known, we manually align the generated mesh to the scene.

8. Ambiguity in Completion

As shown in Fig. 10, object completion is inherently ambiguous. While varying seeds produce different outcomes, the results converge as input views increase.

9. Additional Results of In-situ Generation

We present additional examples of in-situ generation as shown in Fig. 12. We put the reconstructed results from each method back into the original scene to demonstrate the alignment to the real-world scans. Additionally, we showcase the plausibility of the reconstructed unseen regions. Our method recovers the unseen regions while preserving the most realistic rendering quality. We also conduct quantitative comparisons on the user study and demonstrate superior outcomes as shown in Tab. 2. We ask 30 users to sort 20 testing instances in random order based on the alignment with the original scene and the faithful completion of unseen regions from both appearance rendering and geometry, and assign the scores by their ranking (i.e., with a score of 3 for the ordered best one and a score of 1 for the last one) following TEXTure [13].

10. Failure Cases Arising from the Limitations

As mentioned in conclusion, if the scene itself cannot be well reconstructed by Gaussian Splatting (limited viewing angles of the input images or challenging material of the object itself make reconstruction difficult), low-quality novel view rendering of segmented objects will affect the generative model when used as condition images (as shown in Fig. 11). Optimizing GS reconstruction is beyond the scope of our research.

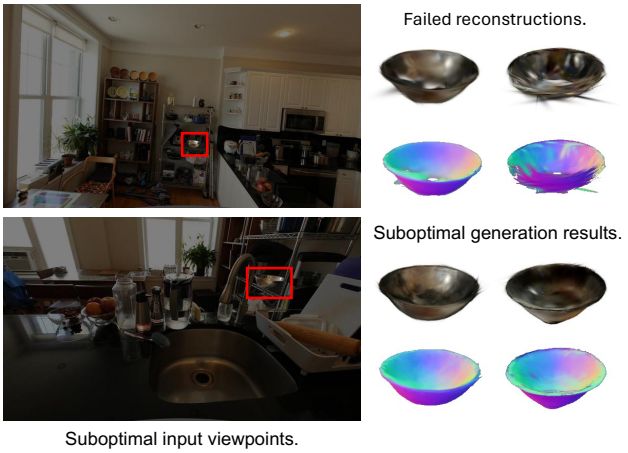


Figure 11. **Failure Cases.** Due to suboptimal input viewpoints (i.e., the visible parts of the object were not sufficiently observed at close range), the instance could not be accurately reconstructed. Consequently, the condition images provided for in-situ generation were of poor quality, ultimately resulting in inferior generation outcomes.

References

[1] Jonathan T Barron, Ben Mildenhall, Dor Verbin, Pratul P Srinivasan, and Peter Hedman. Zip-nerf: Anti-aliased grid-based neural radiance fields. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 19697–19705, 2023. 2

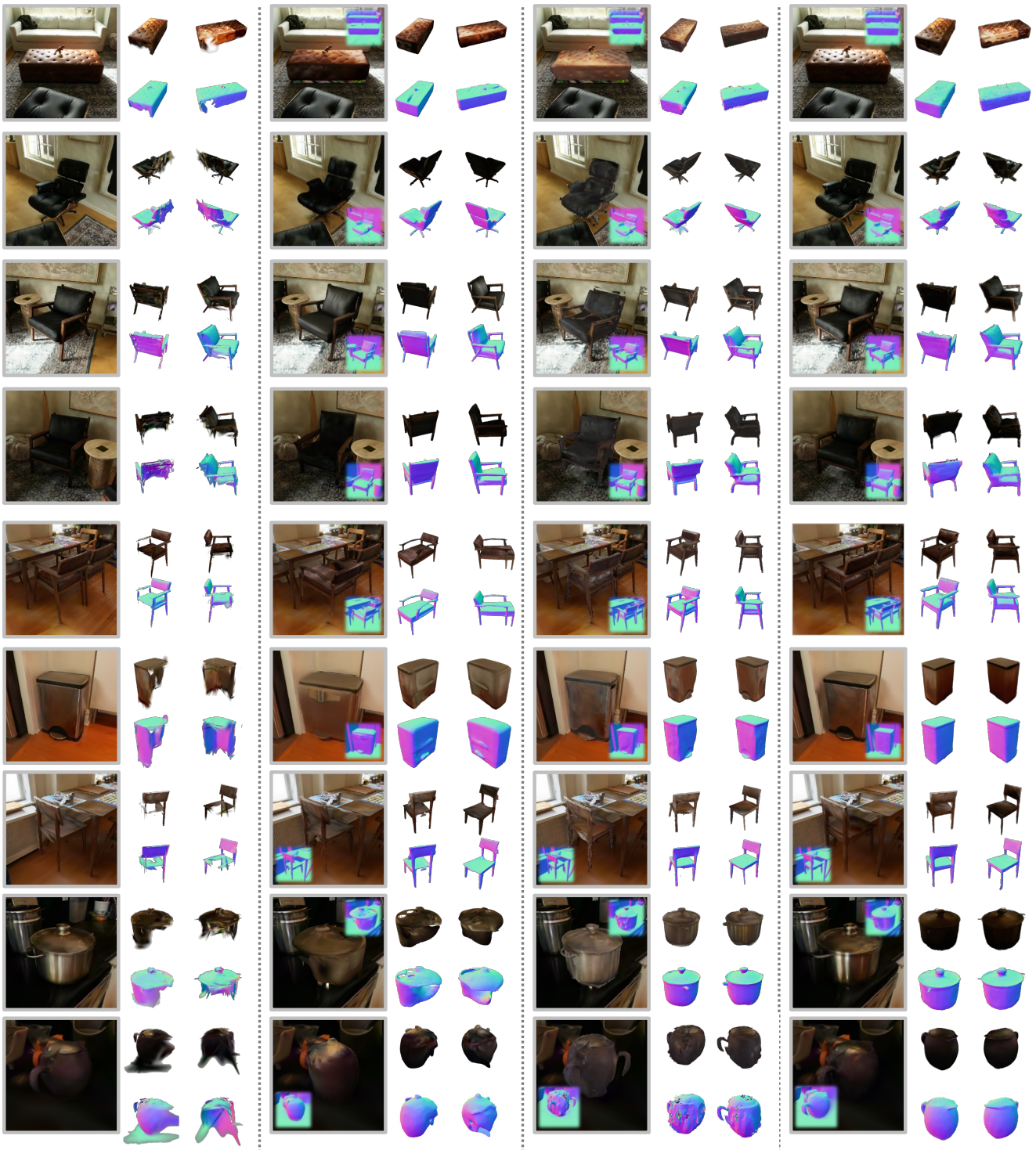
[2] Yiwen Chen, Zilong Chen, Chi Zhang, Feng Wang, Xiaofeng Yang, Yikai Wang, Zhongang Cai, Lei Yang, Huaping Liu, and Guosheng Lin. Gaussianeditor: Swift and controllable 3d editing with gaussian splatting. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 21476–21485, 2024. 4

[3] Matt Deitke, Dustin Schwenk, Jordi Salvador, Luca Weihs, Oscar Michel, Eli VanderBilt, Ludwig Schmidt, Kiana Ehsani, Aniruddha Kembhavi, and Ali Farhadi. Objaverse: A universe of annotated 3d objects. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 13142–13153, 2023. 4

[4] Hanzhe Hu, Zhizhuo Zhou, Varun Jampani, and Shubham Tulsiani. Mvd-fusion: Single-view 3d via depth-consistent multi-view generation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 9698–9707, 2024. 4

[5] 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. 1, 4

[6] Ying Jiang, Chang Yu, Tianyi Xie, Xuan Li, Yutao Feng, Huamin Wang, Minchen Li, Henry Lau, Feng Gao, Yin Yang, et al. Vr-gs: A physical dynamics-aware interactive



(a) Original Scene (b) InstantMesh Rendering (c) SpaRP Rendering (d) Our In-Situ Gen. and Rendering

Figure 12. **Additional visualization of the in-situ generation.** We place the generated object back into the original scene for comparison in terms of both shape and geometric alignment.

gaussian splatting system in virtual reality. In *ACM SIGGRAPH 2024 Conference Papers*, pages 1–1, 2024. 4

[7] 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. 1, 2

[8] 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. 3

[9] Ruoshi Liu, Rundi Wu, Basile Van Hoorick, Pavel Tokmakov, Sergey Zakharov, and Carl Vondrick. Zero-1-to-3: Zero-shot one image to 3d object. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 9298–9309, 2023. 4

[10] Yuan Liu, Cheng Lin, Zijiao Zeng, Xiaoxiao Long, Lingjie Liu, Taku Komura, and Wenping Wang. Syncdreamer: Generating multiview-consistent images from a single-view image. *arXiv preprint arXiv:2309.03453*, 2023. 4

[11] Lu Qi, Jason Kuen, Weidong Guo, Tiancheng Shen, Jiuxiang Gu, Jiaya Jia, Zhe Lin, and Ming-Hsuan Yang. High-quality entity segmentation. *arXiv preprint arXiv:2211.05776*, 2022. 1

[12] Minghan Qin, Wanhua Li, Jiawei Zhou, Haoqian Wang, and Hanspeter Pfister. Langsplat: 3d language gaussian splatting. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 20051–20060, 2024. 3

[13] Elad Richardson, Gal Metzer, Yuval Alaluf, Raja Giryes, and Daniel Cohen-Or. Texture: Text-guided texturing of 3d shapes. In *ACM SIGGRAPH 2023 conference proceedings*, pages 1–11, 2023. 5

[14] Ruoxi Shi, Hansheng Chen, Zhuoyang Zhang, Minghua Liu, Chao Xu, Xinyue Wei, Linghao Chen, Chong Zeng, and Hao Su. Zero123++: a single image to consistent multi-view diffusion base model. *arXiv preprint arXiv:2310.15110*, 2023. 4

[15] Chao Xu, Ang Li, Linghao Chen, Yulin Liu, Ruoxi Shi, Hao Su, and Minghua Liu. Sparp: Fast 3d object reconstruction and pose estimation from sparse views. In *European Conference on Computer Vision*, pages 143–163. Springer, 2025. 4, 5

[16] Jiale Xu, Weihao Cheng, Yiming Gao, Xintao Wang, Shenghua Gao, and Ying Shan. Instantmesh: Efficient 3d mesh generation from a single image with sparse-view large reconstruction models. *arXiv preprint arXiv:2404.07191*, 2024. 4, 5

[17] 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, 2025. 4

[18] Junhao Zhuang, Yanhong Zeng, Wenran Liu, Chun Yuan, and Kai Chen. A task is worth one word: Learning with task prompts for high-quality versatile image inpainting, 2023. 4