Inpaint360GS: Efficient Object-Aware 3D Inpainting via Gaussian Splatting for 360° Scenes

Shaoxiang Wang[1,2] Shihong Zhang[3] Christen Millerdurai[1]
Rüdiger Westermann[3] Didier Stricker[1,2] Alain Pagani[1]

[1]German Research Center for Artificial Intelligence [2]RPTU [3]Technical University of Munich

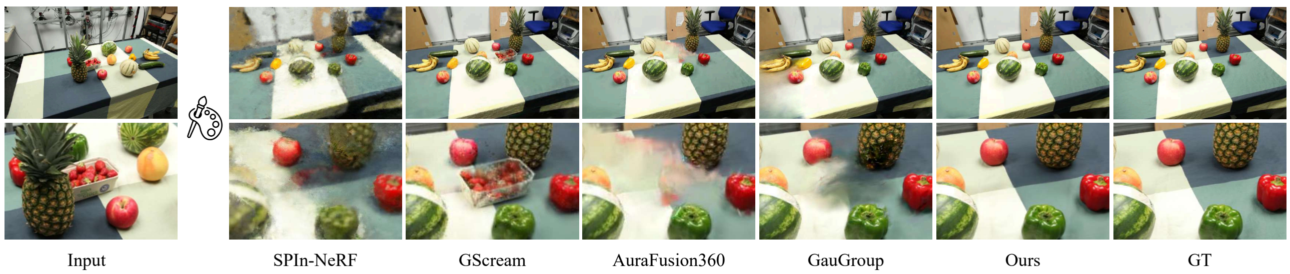

Figure 1. We propose a novel object-aware 3D inpainting method, *Inpaint360GS*, which flexibly enables object removal and inpainting in 360° scenes. Our approach effectively handles occlusions in multi-object environments and achieves better geometric and appearance consistency compared to existing state-of-the-art methods, including SPIn-NeRF [28], GScream [43], AuraFusion360 [48], and GauGroup [53].

Abstract

Despite recent advances in single-object front-facing inpainting using NeRF and 3D Gaussian Splatting (3DGS), inpainting in complex 360° *scenes remains largely underexplored. This is primarily due to three key challenges: (i) identifying target objects in the 3D field of 360° environments, (ii) dealing with severe occlusions in multi-object scenes, which makes it hard to define regions to inpaint, and (iii) maintaining consistent and high-quality appearance across views effectively.*

To tackle these challenges, we propose Inpaint360GS, a flexible 360° *editing framework based on 3DGS that supports multi-object removal and high-fidelity inpainting in 3D space. By distilling 2D segmentation into 3D and leveraging virtual camera views for contextual guidance, our method enables accurate object-level editing and consistent scene completion. We further introduce a new dataset tailored for* 360° *inpainting, addressing the lack of ground truth object-free scenes. Experiments demonstrate that Inpaint360GS outperforms existing baselines and achieves state-of-the-art performance. Project page: https://dfki-av.github.io/inpaint360gs/*

1. Introduction

Recent advances in 3D scene modeling, such as Neural Radiance Fields (NeRFs) [27] and 3D Gaussian Splatting (3DGS) [19], have enabled realistic view synthesis and high-quality reconstruction. However, vanilla versions of these methods are not designed for scene editing tasks [41, 50] such as object removal or inpainting, especially in complex 360° environments with multiple objects and occlusions. Existing approaches often assume front-facing, single-object setups, and struggle with consistent geometry recovery, object segmentation, and multi-view coherence.

Inpainting on 360° 3D scene, this task poses three key challenges: (1) the need for an editable scene representation that supports object segmentation based on flexible prompts (*e.g.*, via VLMs or clicks); (2) defining the underlying never-before-seen (NBS) regions after object removal, especially under occlusion; and (3) ensuring fast and view-consistent inpainting that preserves structural and virtual continuity across multiple views. Addressing these challenges requires a scene representation that is both editable and spatially explicit. While several NeRF-based methods [6, 28, 40, 45] attempt 3D inpainting, the implicit nature of radiance fields lacks explicit spatial boundaries, limiting object-aware editing. By contrast, 3DGS discretizes scenes into explicit Gaussian elements, supporting localized modification. Nevertheless, despite recent advancements in 3DGS-based approaches [29, 43, 53], achieving efficient 360° multi-view consistent 3D object inpainting remains an open challenge. Although some methods [15, 37, 48] consider view consistency, they typically rely on predefined single-object masks and post refinements. These constraints significantly limit their flexibility for interactive multi-object segmentation. Moreover, the long optimization time required by such methods makes rapid scene editing infeasible.

To address these issues, we propose *Inpaint360GS*, a novel framework for multi-object, multi-view consistent inpainting using 3D Gaussian Splatting. We distill 2D segmentation masks into a 3D Gaussian field to assign per-Gaussian object labels. To ensure geometric consistency across views, we leverage the depth information encoded in the Gaussians to guide the inpainting process without requiring explicit depth alignment. This enables fast convergence and high-fidelity results. Unlike prior methods [28, 43, 48, 53] that rely solely on given camera poses, our approach exploits the view synthesis capability of the 3D Gaussian field to generate virtual camera views centered around the removed objects. These virtual views provide enriched contextual information to guide the inpainting process. Finally, to address the lack of datasets, we introduce a new 360° benchmark dataset comprising indoor and outdoor scenes with single/multiple objects, along with corresponding object-free ground-truth sequences for quantitative evaluation.

In summary, our key contributions include:

- A framework for consistent 2D mask association that integrates 2D segmentation masks into the 3D Gaussian scene representation. While existing works often focus on single-object, our method is explicitly designed for inpainting in 3DGS under multi-object scenarios.
- An efficient depth-guided inpainting method that achieves multi-view completion with consistent structure and texture via virtual camera poses.
- A new benchmark dataset featuring 360° indoor and outdoor sequences containing single/multi objects with varying complexity, along with corresponding object-free ground-truth sequences.

2. Related Work

Efficient and flexible object-level 3D inpainting tasks integrate multiple techniques. To highlight our contributions, we focus the related work discussion on segmentation and inpainting methods that are most relevant to this task.

3D Scene Segmentation. Recent advances in segmentation have been led by models such as SAM [21], HQSAM [32], and SEEM [60], which enable zero-shot 2D segmentation. Building on this progress, temporal methods [7, 8, 13, 22, 47] propagate masks across video frames to maintain consistency over time. Meanwhile, fully supervised 3D instance segmentation [35, 36, 38] has shown promise results, but remains constrained by limited annotated data and often lacks explicit object-level representations due to the scarcity of densely annotated 3D datasets.

To achieve spatially coherent segmentation in 3D, several methods distill 2D masks [3, 4, 26, 52] into radiance fields, while others leverage language embeddings to ground semantics directly in 3D, either in Gaussian Splatting [17, 33] or through transformer-based visual grounding models such as MiKASA [5]. However, these methods are computationally intensive and unsuitable for interactive editing. In contrast, approaches like DEVA [7] improve multi-object handling by decoupling per-frame segmentation from temporal association, benefiting better scalability to multi-object scene applications such as semantic SLAM [58] and Gaussian-based modeling [11, 53]. Still, this video-level 2D label propagation often leads to segmentation errors, which degrade downstream tasks like inpainting and editing in GauGroup [53]. To overcome these challenges, our work proposes efficient segmentation association in 3D Gaussian field. By associating raw 2D segmentation outputs and aligning 2D masks in the Gaussian field, we ensure robust multi-view consistency and mitigate the spatial inconsistencies inherent in purely 2D-driven methods as shown in Fig. 2.

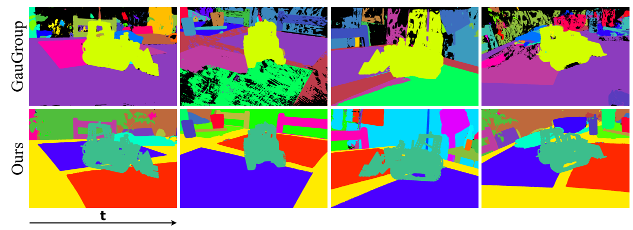

Figure 2. **Multi-View Segmentation Comparison.** Compared to GauGroup [53] our method has more consistent segmentation results across different views.

Inpainting. Classical inpainting methods, such as pixel diffusion [2] and patch-based approaches [9], struggle with large or semantically complex regions. Deep learning introduced generative inpainting using context-encoder GANs [31, 55], though early results were often blurry. Two-stage methods like EdgeConnect [30] improve structure before texture, and recent diffusion-based models [10, 25, 49, 56] offer higher-quality results at significant computational cost. Extending inpainting to 3D requires appropriate scene representations. SPIn-NeRF [28] pioneered front-facing 3D inpainting via implicit fields. However, the more challenging 360° setting demands multi-view consistency, which is hard to achieve with per-view 2D inpainting. NeRF-based methods [6, 40, 45, 46, 54] attempt to integrate multi-view 2D inpainting with 3D optimization, but often suffer from inconsistency due to diffusion output's diversity and geometry misalignment, limiting them to bounded or small-angle scenes [23]. Alternatively, Gaussian-based methods provide explicit scene representations that are inherently more suitable for flexible scene editing. Approaches such as InFusion [24], AuraFusion360 [48], and GScream [43] rely on depth foundation models [18, 51], leading to depth alignment issues. GauGroup [53] injects semantics into Gaussians but remains sensitive to initialization and 2D segmentation quality. Recent works [15, 37, 48] have further improved multi-view consistency and unseen region detection. Nonetheless, these methods struggle with severe occlusions in complex multi-object scenes, and editing remains costly due to depth scale misalignment and localized texture refinement. They also lack of strategies for selecting informative inpainting views, operating only on training poses. To overcome these limitations, we define depth directly from the Gaussian field to eliminate scale ambiguity and introduce a conditional virtual view selection strategy, enabling high-fidelity inpainting and efficient convergence in unbounded 360° environments.

3. Method

We propose an object-aware inpainting framework based on 3DGS. In Sec. 3.1, we review the 3DGS representation. We introduce 2D mask association across views via a Key Object Management System in Gaussian field (Sec. 3.2). These labels are distilled into 3D (Sec. 3.3). After object removal, virtual views are rendered to expose occluded regions (Sec. 3.4). We perform conditional 2D inpainting followed by depth-guided 3D inpainting with hybrid supervision (Sec. 3.5). A new benchmark dataset for 360° inpainting is introduced in Sec. 3.6.

3.1. Preliminaries

3D Gaussian Splatting (3DGS) represents a 3D scene field using a set of Gaussians $G = \{g_i\}_{i=1}^N$ and employs a differentiable rasterizer [19] for efficient rendering, where N is the total number of Gaussians. Each Gaussian $g_i = \{\mathbf{p}_i, \mathbf{s}_i, \mathbf{q}_i, \mathbf{o}_i, \mathbf{c}_i\}$ is defined by its 3D center position $\mathbf{p}_i \in \mathbb{R}^3$, scaling factors $\mathbf{s}_i \in \mathbb{R}^3$, a quaternion $\mathbf{q}_i \in \mathbb{R}^4$ representing 3D orientation and covariance, an opacity value $\mathbf{o}_i \in \mathbb{R}$, and color coefficients $\mathbf{c}_i$ represented using spherical harmonics (SH).

After projecting the 3D Gaussians onto the 2D image plane, 3DGS utilizes the differentiable rasterizer to compute the final pixel color through α-blending of depth-ordered Gaussians. The color $\mathbf{C}$ at a pixel is computed as:

$$\mathbf{C} = \sum_{i \in \mathcal{N}} \mathbf{c}_i \alpha_i T_i, \tag{1}$$

where $\mathcal{N}$ is the set of Gaussians overlapping the pixel, α_i represents the influence of the i-th Gaussian, and T_i is the accumulated transmittance defined as $T_i = \prod_{j=1}^{i-1}(1-\alpha_j)$.

3.2. 2D segmentation mask association via 3D Gaussian

Our 3D scene is represented using Gaussians, as described in Sec. 3.1. To support object-level editing, each Gaussian must be assigned a unique and consistent object ID across views. A naïve approach projects Gaussians onto 2D masks from models like SAM [21], but these masks often produce inconsistent labels across viewpoints. GauGroup [53] tackles this using DEVA [7] to associate object masks across views by treating the image sequence as a video. Specifically, it fails under sparse-view settings.

To address this issue, we introduce the Key Object Management System, a label association mechanism that ensures consistent object ID assignment for 3D Gaussians.

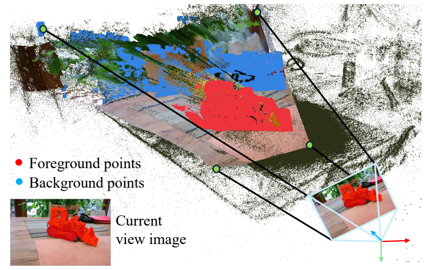

Figure 3. **Projection of 3D Gaussians onto 2D Segmentation.** K-Means algorithm is employed to effectively distinguish between the foreground (*i.e.*, target object) and background Gaussian points.

Fig. 2 shows the resulting ID assignment of our more robust alternative. This mechanism is analogous to the keyframe overlap check used in SLAM systems [16, 42, 59], which measures the shared visible content between frames, but here it is adapted to assign view consistent 2D object labels to 3D Gaussian sets. The Key Object Management System maintains a Key Object Database, denoted as $\mathcal{D}_{\text{ID}}$, which maps object IDs to their corresponding Gaussian sets. Suppose there are Q distinct objects in the scene; then, we define the database as $\mathcal{D}_{\text{ID}} = \{P_1, P_2, \ldots, P_Q\}$, where each P_i represents the set of Gaussians belonging to the i-th object. Specifically, $P_i = \{g_i^1, g_i^2, \ldots, g_i^m\}$, where g_i^k denotes the k-th Gaussian associated with the i-th object and m is the total number of Gaussians for that object.

Key Object Database. To obtain the P_i set of Gaussians belonging to the i-th object, we first project all Gaussians into 2D image coordinates using the corresponding camera poses (see Sec. 3.1). We then assign the 2D object labels to these Gaussians. However, not every projected Gaussian actually belongs to the object. As shown in Fig. 3, only the red points correspond to the truck's foreground, while the blue points belong to the background despite overlapping with the truck's segmentation. To differentiate these, we apply K-Means clustering with $K = 2$ in Euclidean space to partition the Gaussians into foreground (*i.e.*, object) and background groups. We assign 2D segmentation labels only to the Gaussian cluster closer to the camera, ensuring accurate foreground association. This process is repeated across all training views to establish consistent object-Gaussian correspondences.

Key Object Management System. The Key Object Management System is used to merge and create new P_i sets in the Key Object Database to ensure consistent object ID assignments across all frames. For each view, we first assign temporary object IDs to Gaussians based on the 2D segmentation results. Then, the Gaussians g_i associated with each object in the current view are compared with those stored in the Key Object Database $\mathcal{D}_{\text{ID}}$. To perform this comparison, we define the Gaussian Set Intersection-over-Union (GS-IoU) metric to quantify the overlap between Gaussian sets from different views. Specifically, the GS-IoU between the i-th proposal and the

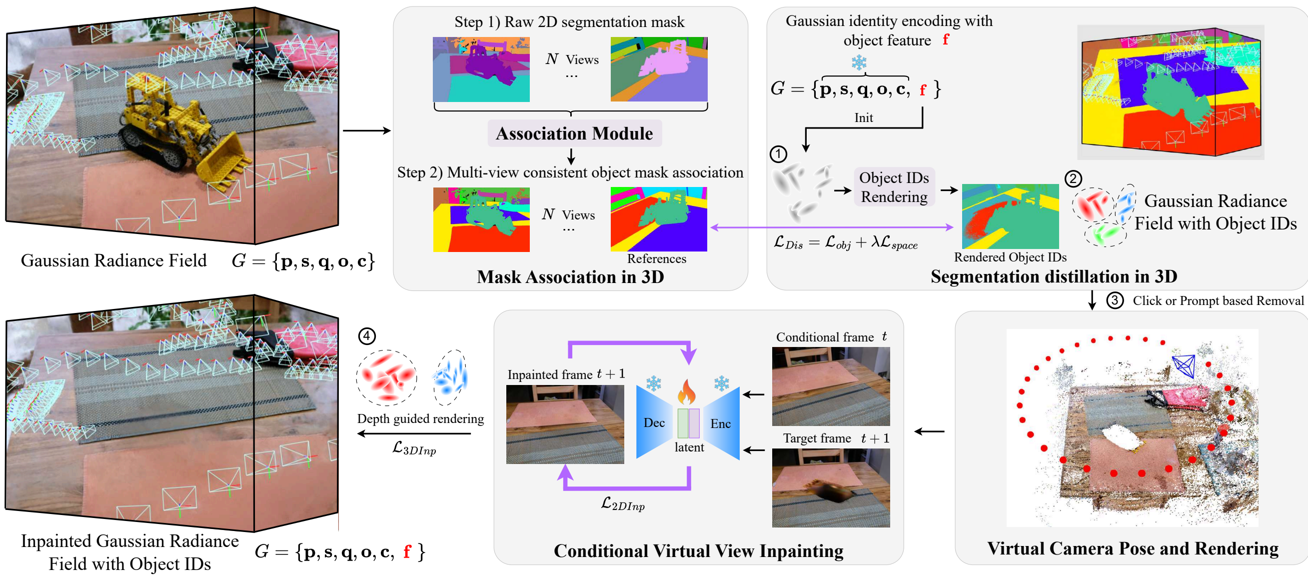

Figure 4. **Inpaint360GS Architecture Overview.** Our framework takes a sequence of RGB images to construct a Gaussian Radiance Field (GRF) and extract per-view object masks using a 2D segmentation foundation model. By associating these masks across views within the GRF, we obtain multi-view consistent object masks and embed them into the Gaussian representation, assigning each Gaussian an object ID. This object-aware GRF enables direct 3D object manipulation, such as click-based or prompt-based removal. After removing target objects, we render at novel camera poses to obtain virtual views $\mathcal{V}$. During 2D inpainting, we recursively perform conditional RGB and depth inpainting, which is then used for depth-guided 3D inpainting.

j-th proposal is defined as:

$$\text{GS-IoU}_{ij} = \frac{|P_i \cap P_j|}{|P_i \cup P_j|} \quad 0 \leq \text{GS-IoU}_{ij} \leq 1, \tag{2}$$

where P_i represents the set of Gaussian indices associated with the i-th object in the current view, and $P_j \in \mathcal{D}_{\text{ID}}$ denotes the set of indices for the j-th object stored in the database. If the GS-IoU exceeds a threshold σ, the object is matched to an existing entry in the database, and its Gaussians inherit the corresponding object ID; otherwise, it is treated as a new instance with a new ID. After processing all views sequentially along a continuous camera poses, the Key Object Database contains roughly labeled Gaussians across viewpoints. We emphasize that these rough ID of Gaussians need not be perfect–they mainly serve to associate raw 2D segmentation masks across views, yielding consistent object masks O that are later used as ground truth for the object ID distillation stage. For further corner case (bird-view dense objects scenariom, sparse view case) analysis and visualization results, please refer to Supp. Sec. 3.

3.3. Efficient Object ID Distillation in 3D

Directly mapping object IDs from the Key Object Database often yields noisy or incomplete labels, resulting in unreliable point clouds. To address this, we distill the associated object mask from Key Object Database, ensuring consistency across views.

We distill the 2D object masks into the Gaussian field following the approach of GauGroup [53]. Each Gaussian point is associated with a randomly initialized feature vector f that represents its object ID embedding. Next, we apply α-blending to obtain a feature map:

$$\mathbf{F} = \sum_{i \in \mathcal{N}} f_i \alpha_i T_i, \tag{3}$$

where α_i denotes the influence of the i-th Gaussian, and T_i represents the transmittance. Subsequently, a linear transformation $\Phi(\cdot)$ projects the feature dimension to Q, corresponding to the total quantity of distinct objects in the scene. The resulting feature vectors are then processed with a *softmax* for identity classification, i.e., $\hat{O} = \text{softmax}(\Phi(F))$, where $\hat{O} \in \mathbb{R}^{H \times W \times Q}$, with H and W representing the height and width of the image respectively. We compute the 2D classification loss using the multi-class cross-entropy, i.e., $\mathcal{L}_{obj} = \text{CrossEntropy}(\hat{O}, O)$, where $O \in \mathbb{R}^{H \times W \times Q}$ is the associated 2D object mask with Q classes (see Sec. 3.2). Additionally, we introduce 3D spatial supervision loss to complement the 2D supervision, which significantly accelerates convergence and enables more efficient distillation, particularly around complex object boundaries and fine structures. For a given Gaussian point with feature vector f_i, we consider its k-nearest neighbors $\mathcal{K}(f_i) = \{f_i^1, f_i^2, \ldots, f_i^k\}$ in Euclidean space and encourage these neighboring features to be similar. We then define the *spatial similarity loss* between f_i and its neighbors as:

$$\mathcal{L}_{space} = 1 - \sum_{i \in k} \frac{f_i \cdot f_k}{\|f_i\| \|f_k\|}. \tag{4}$$

The overall loss function for this distillation process is then given by

$$\mathcal{L}_{Dis} = \mathcal{L}_{obj} + \lambda \mathcal{L}_{space}, \tag{5}$$

where λ is a balancing factor that regulates the contribution of the spatial consistency loss to the total loss.

3.4. Virtual Camera Views for Inpainting

With each Gaussian assigned a unique object ID, object removal via clicks or prompts becomes straightforward. After removal, only occluded or never-before-seen (NBS) regions(*e.g.*, the base of the object), as most background areas remain visible from other views. Accurately identifying minimal NBS regions preserves valid content and reduces unnecessary inpainting. Prior work [53] uses SAM-Tracking (SAMT) [8] to detect NBS regions, but it fails under discontinuous frames. Recent methods [15, 37] rely on iterative 3D-to-2D projections and learnable masks, but they are often inaccurate, computationally expensive, and limited to single-object scenarios.

In contrast, our method fully leverages the 3D Gaussian field's capability to synthesize novel views. We apply PCA-based pose alignment to generate a virtual circular trajectory centered on the removed object. Given an optimized 3D Gaussian scene $\mathcal{R}$, we first compute the object center and define a virtual trajectory $\mathcal{P} = \{p_i\}_{i=1}^{L}$ based on the original camera poses, where p_i denotes the pose at frame i and L is the total number of views. For each pose p_i, we render an RGB image C_i and its corresponding depth map D_i.

$$\mathcal{V} = \big\{(C_i, D_i, M_i) \,\big|\, (C_i, D_i) = \texttt{render}(p_i, \mathcal{R}), \\ M_i = \texttt{SAMT}(C_i)\big\}_{i=1}^{L} \tag{6}$$

For multi-object scenes, object occlusion could be addressed by leveraging lightweight object detectors (*e.g.*, YOLOv8 [34]) to identify overlapping instances. Occluding objects around the target are temporarily removed to facilitate reliable NBS region mask M_i extraction using SAMT [8], enabled by the smooth viewpoint transitions. The trajectory radius is adaptively controlled to ensure sufficient coverage of occluded regions without introducing extreme viewpoints. As a result, we obtain a set of virtual views $\mathcal{V} = \{(C_i, D_i, M_i)\}_{i=1}^{L}$, which serve as input for the inpainting stage.

3.5. Depth Guided Multi-view Consistent Inpainting

We address three key challenges in this module: (1) inpainting never-before-seen (NBS) regions on 2D images, (2) initializing inpainted content directly on the 3D scene surface for efficient integration, and (3) optimizing the inpainting process in 3D space.

Recursive Conditional Inpainting. A major challenge in achieving 360° coherent rendering of the 3D Gaussian field lies in maintaining multi-view consistency during inpainting. Prior methods [28, 43, 48, 53] are limited to fixed training camera views. For extreme viewpoints(*e.g.*, oblique angles or views with very small NBS regions) 2D inpainting often results in poor textures and noticeable artifacts.

To overcome this, we leverage a set of continuous virtual frames $\mathcal{V}$. To avoid hallucination artifacts, we adopt Fourier convolutions LaMa [39] as the inpainting model to fill the removed regions in the first virtual frame. Starting from the second frame, we use the previously inpainted frame as a conditional reference to guide the inpainting of the current frame. This recursive process ensures that each frame's texture is guided by the previous one, thereby maintaining temporal and visual consistency. Specifically, both the inpainted frame C_t and the target frame C_{t+1} are encoded into a shared latent space via a conditional encoder: $\ell_t, \ell_{t+1} = \text{Encoder}(C_t, C_{t+1})$. The resulting latent features are concatenated and optimized jointly in the feature space. The inpainting loss is defined as:

$$\mathcal{L}_{\text{2DInp}} = \left\| (C_t - \hat{C}_t) \right\|_1 + \left\| M_{t+1} \odot (C_{t+1} - \hat{C}_{t+1}) \right\|_1. \tag{7}$$

The corresponding mask M_{t+1} indicates the regions to be filled. $\hat{C}_t, \hat{C}_{t+1}$ are decoded image after every step. After 10 optimization steps, the completed image is obtained by decoding the updated feature: $C_{t+1} = \text{Decoder}(\ell_t, \ell_{t+1})$. This strategy effectively overcomes the issue of view discontinuity in the training dataset. Moreover, the recursive conditional guidance enforces temporal continuity of texture information, fully leveraging the capability of novel view synthesis in 3D.

Depth-Guided Gaussian Initialization. Initializing the 3D point cloud is critical for successful Gaussian Splatting reconstruction. While Infusion [24] relies on a depth completion model, AuraFusion360 [48] and GScream [43] adopt Marigold [18] for zero-shot depth estimation followed by scale alignment via diffusion models. However, these methods introduce additional dependencies and substantially increase training time.

Instead, we leverage the intrinsic properties of the Gaussian field to define the depth as:

$$\mathbf{D} = \sum_{i \in \mathcal{N}} z_i \alpha_i T_i, \tag{8}$$

where z_i is the z-coordinate of the i-th Gaussian in the camera coordinate system, α_i denotes the influence of i-th Gaussian, and T_i is the accumulated transmittance. Since missing depth regions typically exhibit low texture complexity than color image, they can be effectively inpainted using models like LaMa [39]. Given the inpainted depth D_{inp} and the corresponding color image C_{inp}, we fuse them with the inpainting mask M to obtain a point cloud for the NBS region, which is then used to initialize the Gaussians.

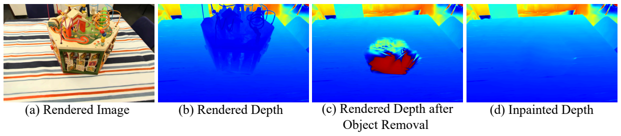

Figure 5. **Depth Completion.** Leveraging the inherent structure of the scene, our method performs depth inpainting without requiring explicit depth alignment.

3D Inpainting. During the 3D scene inpainting phase, an intuitive idea is to make the Gaussians in the remaining (non-masked) regions non-trainable, and optimize only those within the masked areas. However, empirical observations indicate that this strategy tends to produce noisy textures and unstable boundary transitions.To address this, we propose a *3D hybrid supervision scheme* that combines localized and global objectives. Specifically, we supervise masked regions using $\mathcal{L}_1$ and $\mathcal{L}_{\text{LPIPS}}$ losses, while enforcing global structural consistency with SSIM computed over the entire image:

$$\mathcal{L}_{\text{3DInp}} = (1-\lambda_1)\left\| M \odot (C_{inp} - \hat{C}) \right\|_1 + \lambda_1 \mathcal{L}_{\text{D-SSIM}}(C_{inp}, \hat{C}) + \lambda_2 \mathcal{L}_{\text{LPIPS}}(C_{inp}, \hat{C}, M). \tag{9}$$

Here, M denotes the binary inpainting mask, $\hat{C}$ the rendered image, and C_{inp} the inpainted result used as reference. Unlike SSIM, which is sensitive to localized inconsistencies when computed within small masks, applying it over the full image stabilizes optimization and improves boundary smoothness.

3.6. Dataset for 360° Inpainting

Existing radiance field datasets are unsuitable for 360° inpainting due to several limitations. Datasets like NeRF2NeRF [12], MipNeRF360 [1], and LERF [20] lack object-free(without object) scenes, making quantitative evaluation infeasible. While SPIn-NeRF [28] offers object-free ground truth, it is limited to front-facing views and indoor scenes, with photometric inconsistencies caused by varying camera settings. Other datasets [37, 48] lack multi-object scenarios and suffer from test-view leakage in point clouds, further undermining the validity of quantitative evaluations. To address these issues, we introduce a new 360° inpainting dataset with object-inclusive and object-free sequences. It contains 11 scenes: 7 single-object and 4 multi-object settings with occlusions, covering diverse indoor and outdoor environments. Camera parameters (exposure, white balance, ISO) are fixed to eliminate photometric variation. To ensure fair evaluation, test-view point clouds are excluded from training. See Supp. Sec. 1 for details.

4. Experiments and Results

4.1. Experimental setup.

Datasets. We evaluate Inpaint360GS across multiple benchmarks: (1) Inpaint360GS (ours): A new dataset with 11 scenes (7 single-object, 4 multi-object). All experiments are conducted at $1/4$ resolution, and evaluations are performed on object-free test images. (2) Additional benchmarks: To demonstrate scalability, we test on three extra scenes collected from Mip-NeRF 360 [1], Instruct-NeRF2NeRF [12], and LERF [20].

Metrics. We evaluate visual quality using PSNR, SSIM [44], LPIPS [57], and Frechet Inception Distance (FID) [14]. All metrics are computed on both full images and NBS region in the Inpaint360GS test set. For external datasets lacking object-free ground truth, we provide qualitative comparisons.

Baselines and Implementation. We compare our methods with four recent baseline methods: SPIn-NeRF [28], GScream [43], AuraFusion360 [48] and GauGroup [53]. We retrain and test the model using their open-source code. All experiments are conducted on a single NVIDIA H100 GPU. For more implementation details, please refer to Supp. Sec. 2.

4.2. Evaluation against State-of-the-Art Methods

Qualitative comparisons. Results on the Inpaint360GS dataset are shown in Fig. 1 and Fig. 6. Our method demonstrates superior texture quality and achieves the best FID score, which highlights the effectiveness of our pipeline design. The virtual camera poses enable accurate identification of NBS regions, while the conditional virtual view inpainting ensures consistent texture generation across multiple views. In Fig. 7, we present inpainting results on the `bear` and `kitchen` scenes. The rightmost column provides a reference image containing the target object. Compared with other baselines, our method achieves noticeably smoother boundaries and more plausible texture synthesis. We attribute this to our conditional inpainting guided by virtual camera poses.

Quantitative Evaluation. In Tab. 1, we report PSNR, SSIM, LPIPS, and FID metrics on the Inpaint360GS dataset for both masked regions and full images. Our method consistently outperforms all baselines across all metrics. Front-facing inpainting baselines like SPIn-NeRF [28] and GScream [43] are fundamentally limited in 360° inpainting due to their lack of multi-view awareness. Although AuraFusion360 [48] targets 360° scenes, it struggles in complex multi-object scenarios, where depth misalignments arise from unreliable NBS region identification as illustrated in Fig. 1. GauGroup [53] supports multi-view inputs, but inconsistent object IDs hinder reliable object removal. In contrast, our method demonstrates robust performance in both single-object and multi-object scenarios. This robustness is primarily attributed to the consistent object IDs maintained across views, which enable reliable cross-view reasoning. Additionally, the use of virtual camera poses allows accurate localization of the NBS

Methods	PSNR ↑	masked PSNR ↑	SSIM ↑	masked SSIM↑	LPIPS ↓	masked LPIPS ↓	FID ↓
SPIn-NeRF [28]	19.71	34.53	0.5000	0.9854	0.5002	0.0140	229.95
GScream [43]	20.95	28.47	0.7380	0.9819	0.2715	0.0161	206.25
AuraFusion360 [48]	23.15	35.78	0.7923	0.9872	0.1915	0.0097	47.71
GauGroup [53]	23.20	35.73	0.7928	0.9862	0.1770	0.0102	65.87
Inpaint360GS (Ours)	**24.40**	**36.29**	**0.8370**	**0.9886**	**0.1300**	**0.0078**	**35.93**

Table 1. **Quantitative comparison of 360° inpainting methods on the Inpaint360GS dataset.**

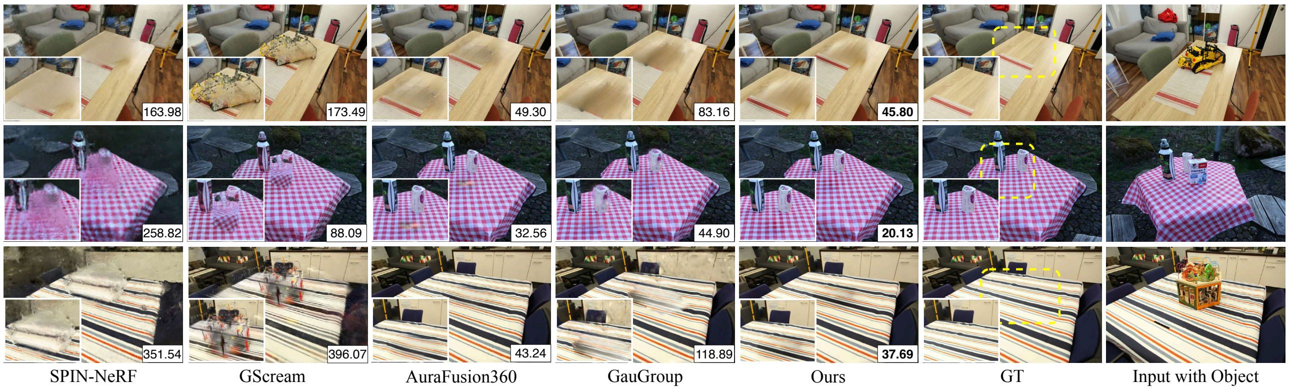

Figure 6. **Inpainting Result Comparison on our Inpaint360GS dataset.** We compare our method with the single-view inpainting approach GScream [43] and the multi-view inpainting methods SPIn-NeRF [28],AuraFusion360 [48] and GauGroup [53]. The metric FID is reported at the right corner. Our approach achieves superior inpainting performance across various scenarios. Please zoom in for details. For per-scene multi-view results, please refer to Supp. Sec. 4.

regions, while the conditional virtual view inpainting effectively enforces multi-view consistency.

In addition, Tab. 2 summarizes the runtime and memory consumption of all methods, evaluated on an NVIDIA H100 GPU using the `kitchen` scene from Mip-NeRF 360 [1] and the `bear` scene from Instruct-NeRF2NeRF [12]. For ours and GauGroup, the vanilla model includes object ID information. The reported inpainting time accounts for both 2D and 3D inpainting stages. In terms of efficiency, our method exhibits two major advantages. First, it maintains consistent object identities across views, which facilitates flexible scene editing. Compared to GauGroup, our approach achieves higher rendering quality while using a model that is 30% more compact. Second, the inpainting stage is 5-10× faster than existing SOTA methods, enabling interactive usage. This efficiency is primarily attributed to accurate depth estimation, which removes the need for explicit alignment, and significantly accelerates the 3D inpainting process. Detailed runtime analysis can be found in Supp. Sec. 3.

4.3. Design Choice and Ablation Study

Effectiveness of Object Mask Association. We compare our object mask association strategy with that of DEVA [7], which is adopted by GauGroup [53]. As shown in Tab. 3 a), using DEVA-generated masks for scene reconstruction results in noticeably degraded performance. In Fig. 2, we provide qualitative evidence of the robustness and cross-view consistency of our method. Moreover, we validate the reliability of our mask association under challenging scenarios, including densely packed objects, bird's-eye viewpoints, and sparse input configurations. Additional visualizations and analyses are provided in Supp. Sec. 3.

Scene	Method	Object ID	Vanilla model training↓/Mins	Inpainting time↓ Mins	Total Time↓ Mins	Storage↓ MB
bear[12]	SPIn-NeRF [28]	✗	79	196	275	336
	GScream [43]	✗	–	52	52	**73.2**
	AuraFusion [48]	✗	25	26	51	448.9
	GauGroup [53]	✓	55	20	75	774.8
	Ours	✓	**21.5**	**2.5**	**24**	477.5
kitchen[1]	SPIn-NeRF [28]	✗	59	148	207	336
	GScream [43]	✗	–	30	30	**67.9**
	AuraFusion [48]	✗	20	43	63	183.5
	GauGroup [53]	✓	27	13	40	897.4
	Ours	✓	**12**	**3**	**15**	663.5

Table 2. **Runtime and Model Size Comparison.** All unnecessary intermediate outputs are disabled to ensure fair comparison across methods.

Method	PSNR	SSIM	LPIPS	FID
a) w/o obj. association	23.31	0.7921	0.1932	66.75
b) w/o depth guidance	24.15	0.8199	0.1256	35.75
c) w/o virtual camera pose	24.18	0.7987	0.1574	38.74
d) w/o cond. inpainting	24.23	0.8156	0.1420	37.57
e) w/o 3D hyb. supervison	24.01	0.7997	0.1398	38.42
f) Ours	**24.40**	**0.8370**	**0.1300**	**33.93**

Table 3. **Ablation on Inpaint360GS dataset.**

Effectiveness of Depth Guidance. Depth guidance substantially contributes to the efficiency of the 3D inpainting process. As reported in Tab. 3 b), removing depth guidance leads to a noticeable decline in reconstruction quality. This highlights the importance of accurate geometric priors in accelerating convergence and enhancing final performance.

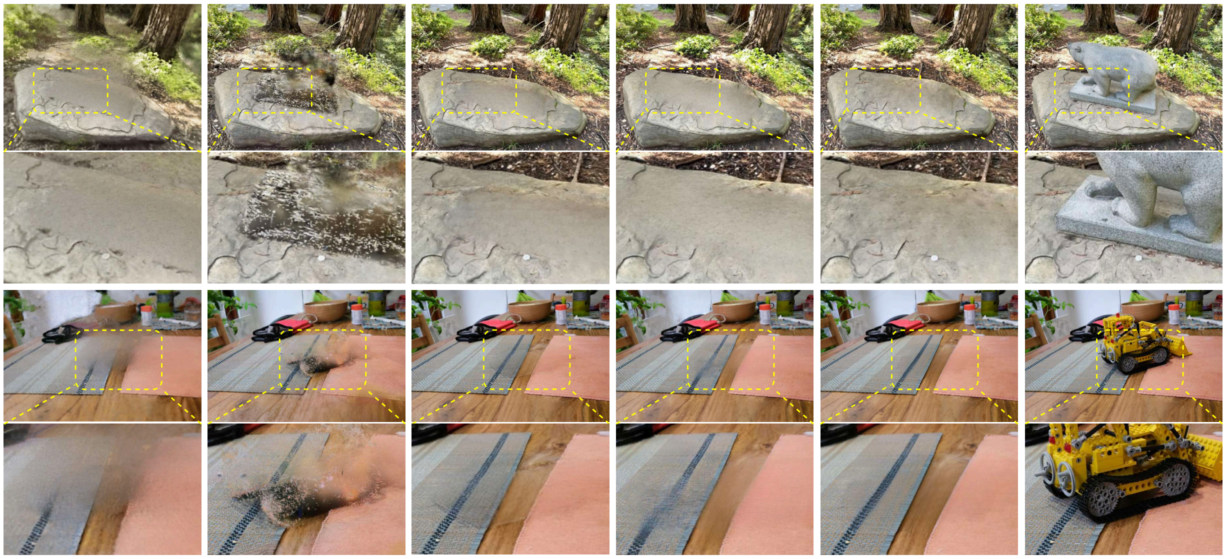

Figure 7. **Inpainting Result Comparison on Instruct-NeRF2NeRF [12] and Mip-NeRF 360 [1].** Our method produces visually plausible 3D inpainted textures with smooth and coherent boundaries.

Effectiveness of Virtual Camera Pose. Virtual camera poses help mitigate the challenges introduced by extreme viewpoints in the training data. In Fig. 8, we demonstrate the detected NBS region. Moreover, our method leverages flexible object identity to perform occlusion-aware inpainting. Specifically, we detect occluded instances using object detection and temporarily remove them before inpainting. After the inpainting process, the temporarily removed objects are reinserted into the scene. This strategy allows the system to better exploit contextual information from the surrounding environment. Tab. 3 c) shows the ablation on it.

Figure 8. **Ablation on virtual camera view.** Compared to the original training views, virtual camera views provide better visibility for NBS regions by overcoming the limitations of extreme viewing angles and occlusions.

Effectiveness of Conditional Inpainting. We adopt a conditional inpainting strategy in which each 2D inpainting step is guided recursively by the previously rendered frame. The use of continuous camera poses facilitates the propagation of consistent texture context across views. As shown in Tab. 3 d) and Fig. 9, removing this strategy leads to a noticeable decline in inpainting quality.

Effectiveness of 3D hybrid supervision. In Tab. 3 e) and Fig. 10, we show that employing the proposed 3D hybrid supervision significantly improves inpainting quality compared to the naive masking-only strategy.

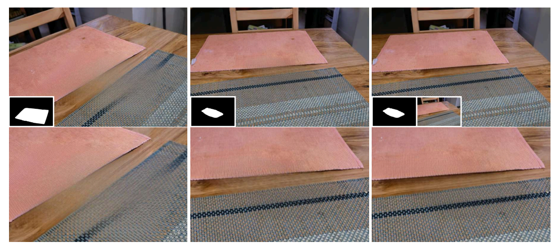

Figure 9. **Ablation on conditional inpainting.**

Figure 10. **Ablation on 3D Hybrid Supervision.**

5. Conclusion

Inpaint360GS is a novel object-aware inpainting framework based on 3D Gaussian Splatting in 360° scenes. By distilling 2D segmentation masks into 3D space and leveraging a virtual-view, depth-guided inpainting strategy, our method enables faster convergence while ensuring structural and photometric consistency. We also introduce a benchmark dataset for 360° inpainting with object-inclusive and object-free data, and extensive experiments show that Inpaint360GS significantly outperforms SOTA methods. Despite these promising results, our approach occasionally exhibits residual shadow artifacts cast by the removed objects and struggles with inpainting irregular complex textures, which remain to be explored in future work.

Acknowledgements: This work has been partially supported by the EU projects CORTEX2 (GA No. 101070192) and LUMINOUS (GA No. 101135724), as well as the project ARROW by the German Research Foundation (DFG, GA No. 564809505).

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. 6, 7, 8

[2] Marcelo Bertalmio, Guillermo Sapiro, Vincent Caselles, and Coloma Ballester. Image inpainting. In *Proceedings of the 27th annual conference on Computer graphics and interactive techniques*, pages 417–424, 2000. 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. 2

[4] Jiazhong Cen, Zanwei Zhou, Jiemin Fang, Wei Shen, Lingxi Xie, Dongsheng Jiang, Xiaopeng Zhang, Qi Tian, et al. Segment anything in 3d with nerfs. *Advances in Neural Information Processing Systems*, 36:25971–25990, 2023. 2

[5] Chun-Peng Chang, Shaoxiang Wang, Alain Pagani, and Didier Stricker. Mikasa: Multi-key-anchor & scene-aware transformer for 3d visual grounding. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 14131–14140, 2024. 2

[6] Honghua Chen, Chen Change Loy, and Xingang Pan. Mvip-nerf: Multi-view 3d inpainting on nerf scenes via diffusion prior. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 5344–5353, 2024. 1, 2

[7] Ho Kei Cheng, Seoung Wug Oh, Brian Price, Alexander Schwing, and Joon-Young Lee. Tracking anything with decoupled video segmentation. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 1316–1326, 2023. 2, 3, 7

[8] Yangming Cheng, Liulei Li, Yuanyou Xu, Xiaodi Li, Zongxin Yang, Wenguan Wang, and Yi Yang. Segment and track anything. *arXiv preprint arXiv:2305.06558*, 2023. 2, 5

[9] Antonio Criminisi, Patrick Pérez, and Kentaro Toyama. Region filling and object removal by exemplar-based image inpainting. *IEEE Transactions on image processing*, 13(9):1200–1212, 2004. 2

[10] Asya Grechka, Guillaume Couairon, and Matthieu Cord. Gradpaint: Gradient-guided inpainting with diffusion models. *Computer Vision and Image Understanding*, 240: 103928, 2024. 2

[11] Jun Guo, Xiaojian Ma, Yue Fan, Huaping Liu, and Qing Li. Semantic gaussians: Open-vocabulary scene understanding with 3d gaussian splatting. *arXiv preprint arXiv:2403.15624*, 2024. 2

[12] Ayaan Haque, Matthew Tancik, Alexei A Efros, Aleksander Holynski, and Angjoo Kanazawa. Instruct-nerf2nerf: Editing 3d scenes with instructions. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 19740–19750, 2023. 6, 7, 8

[13] Miran Heo, Sukjun Hwang, Jeongseok Hyun, Hanjung Kim, Seoung Wug Oh, Joon-Young Lee, and Seon Joo Kim. A generalized framework for video instance segmentation. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 14623–14632, 2023. 2

[14] Martin Heusel, Hubert Ramsauer, Thomas Unterthiner, Bernhard Nessler, and Sepp Hochreiter. Gans trained by a two time-scale update rule converge to a local nash equilibrium. *Advances in neural information processing systems*, 30, 2017. 6

[15] Sheng-Yu Huang, Zi-Ting Chou, and Yu-Chiang Frank Wang. 3d gaussian inpainting with depth-guided cross-view consistency. In *Proceedings of the Computer Vision and Pattern Recognition Conference*, pages 26704–26713, 2025. 1, 2, 5

[16] Mohammad Mahdi Johari, Camilla Carta, and François Fleuret. Eslam: Efficient dense slam system based on hybrid representation of signed distance fields. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 17408–17419, 2023. 3

[17] Kim Jun-Seong, Kim GeonU, Kim Yu-Ji, Yu-Chiang Frank Wang, Jaesung Choe, and Tae-Hyun Oh. Dr. splat: Directly referring 3d gaussian splatting via direct language embedding registration. In *CVPR*, 2025. 2

[18] Bingxin Ke, Anton Obukhov, Shengyu Huang, Nando Metzger, Rodrigo Caye Daudt, and Konrad Schindler. Repurposing diffusion-based image generators for monocular depth estimation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2024. 2, 5

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

[20] Justin* Kerr, Chung Min* Kim, Ken Goldberg, Angjoo Kanazawa, and Matthew Tancik. Lerf: Language embedded radiance fields. In *International Conference on Computer Vision (ICCV)*, 2023. 6

[21] 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 *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 4015–4026, 2023. 2, 3

[22] Minghan Li, Shuai Li, Xindong Zhang, and Lei Zhang. Univs: Unified and universal video segmentation with prompts as queries. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 3227–3238, 2024. 2

[23] Chieh Hubert Lin, Changil Kim, Jia-Bin Huang, Qinbo Li, Chih-Yao Ma, Johannes Kopf, Ming-Hsuan Yang, and Hung-Yu Tseng. Taming latent diffusion model for neural radiance field inpainting. In *European Conference on Computer Vision (ECCV)*, 2024. 2

[24] Zhiheng Liu, Hao Ouyang, Qiuyu Wang, Ka Leong Cheng, Jie Xiao, Kai Zhu, Nan Xue, Yu Liu, Yujun Shen, and Yang Cao. Infusion: Inpainting 3d gaussians via learning depth completion from diffusion prior. *arXiv preprint arXiv:2404.11613*, 2024. 2, 5

[25] Andreas Lugmayr, Martin Danelljan, Andres Romero, Fisher Yu, Radu Timofte, and Luc Van Gool. Repaint: Inpainting using denoising diffusion probabilistic models. In *Proceedings of the IEEE/CVF conference on computer*

vision and pattern recognition, pages 11461–11471, 2022. 2

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

[27] 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*, 65(1):99–106, 2021. 1

[28] Ashkan Mirzaei, Tristan Aumentado-Armstrong, Konstantinos G Derpanis, Jonathan Kelly, Marcus A Brubaker, Igor Gilitschenski, and Alex Levinshtein. Spin-nerf: Multiview segmentation and perceptual inpainting with neural radiance fields. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 20669–20679, 2023. 1, 2, 5, 6, 7

[29] Ashkan Mirzaei, Riccardo De Lutio, Seung Wook Kim, David Acuna, Jonathan Kelly, Sanja Fidler, Igor Gilitschenski, and Zan Gojcic. Reffusion: Reference adapted diffusion models for 3d scene inpainting. *arXiv preprint arXiv:2404.10765*, 2024. 1

[30] Kamyar Nazeri, Eric Ng, Tony Joseph, Faisal Qureshi, and Mehran Ebrahimi. Edgeconnect: Structure guided image inpainting using edge prediction. In *Proceedings of the IEEE/CVF international conference on computer vision workshops*, pages 0–0, 2019. 2

[31] Deepak Pathak, Philipp Krahenbuhl, Jeff Donahue, Trevor Darrell, and Alexei A Efros. Context encoders: Feature learning by inpainting. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 2536–2544, 2016. 2

[32] Lu Qi, Jason Kuen, Tiancheng Shen, Jiuxiang Gu, Weidong Guo, Jiaya Jia, Zhe Lin, and Ming-Hsuan Yang. High-quality entity segmentation. In *ICCV*, 2023. 2

[33] 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. 2

[34] Dillon Reis, Jordan Kupec, Jacqueline Hong, and Ahmad Daoudi. Real-time flying object detection with yolov8. *arXiv preprint arXiv:2305.09972*, 2023. 5

[35] David Rozenberszki, Or Litany, and Angela Dai. Language-grounded indoor 3d semantic segmentation in the wild. In *European Conference on Computer Vision*, pages 125–141. Springer, 2022. 2

[36] David Rozenberszki, Or Litany, and Angela Dai. Unscene3d: Unsupervised 3d instance segmentation for indoor scenes. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2024. 2

[37] Zhihao Shi, Dong Huo, Yuhongze Zhou, Yan Min, Juwei Lu, and Xinxin Zuo. Imfine: 3d inpainting via geometry-guided multi-view refinement. In *Proceedings of the Computer Vision and Pattern Recognition Conference*, pages 26694–26703, 2025. 1, 2, 5, 6

[38] Jiahao Sun, Chunmei Qing, Junpeng Tan, and Xiangmin Xu. Superpoint transformer for 3d scene instance segmentation. In *Proceedings of the AAAI Conference on Artificial Intelligence*, pages 2393–2401, 2023. 2

[39] Roman Suvorov, Elizaveta Logacheva, Anton Mashikhin, Anastasia Remizova, Arsenii Ashukha, Aleksei Silvestrov, Naejin Kong, Harshith Goka, Kiwoong Park, and Victor Lempitsky. Resolution-robust large mask inpainting with fourier convolutions. In *Proceedings of the IEEE/CVF winter conference on applications of computer vision*, pages 2149–2159, 2022. 5

[40] Dongqing Wang, Tong Zhang, Alaa Abboud, and Sabine Süsstrunk. Innerf360: Text-guided 3d-consistent object inpainting on 360-degree neural radiance fields. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 12677–12686, 2024. 1, 2

[41] Junjie Wang, Jiemin Fang, Xiaopeng Zhang, Lingxi Xie, and Qi Tian. Gaussianeditor: Editing 3d gaussians delicately with text instructions. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 20902–20911, 2024. 1

[42] Shaoxiang Wang, Yaxu Xie, Chun-Peng Chang, Christen Millerdurai, Alain Pagani, and Didier Stricker. Uni-slam: Uncertainty-aware neural implicit slam for real-time dense indoor scene reconstruction. *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, 2025. 3

[43] Yuxin Wang, Qianyi Wu, Guofeng Zhang, and Dan Xu. Learning 3d geometry and feature consistent gaussian splatting for object removal. In *European Conference on Computer Vision*, pages 1–17. Springer, 2024. 1, 2, 5, 6, 7

[44] Zhou Wang, Alan C Bovik, Hamid R Sheikh, and Eero P Simoncelli. Image quality assessment: from error visibility to structural similarity. *IEEE transactions on image processing*, 13(4):600–612, 2004. 6

[45] Ethan Weber, Aleksander Holynski, Varun Jampani, Saurabh Saxena, Noah Snavely, Abhishek Kar, and Angjoo Kanazawa. Nerfiller: Completing scenes via generative 3d inpainting. In *CVPR*, 2024. 1, 2

[46] Silvan Weder, Guillermo Garcia-Hernando, Áron Monszpart, Marc Pollefeys, Gabriel Brostow, Michael Firman, and Sara Vicente. Removing objects from neural radiance fields. In *CVPR*, 2023. 2

[47] Yuetian Weng, Mingfei Han, Haoyu He, Mingjie Li, Lina Yao, Xiaojun Chang, and Bohan Zhuang. Mask propagation for efficient video semantic segmentation. *Advances in Neural Information Processing Systems*, 36: 7170–7183, 2023. 2

[48] Chung-Ho Wu, Yang-Jung Chen, Ying-Huan Chen, Jie-Ying Lee, Bo-Hsu Ke, Chun-Wei Tuan Mu, Yi-Chuan Huang, Chin-Yang Lin, Min-Hung Chen, Yen-Yu Lin, et al. Aurafusion360: Augmented unseen region alignment for reference-based 360deg unbounded scene inpainting. In *Proceedings of the Computer Vision and Pattern Recognition Conference*, pages 16366–16376, 2025. 1, 2, 5, 6, 7

[49] Shaoan Xie, Zhifei Zhang, Zhe Lin, Tobias Hinz, and Kun Zhang. Smartbrush: Text and shape guided object inpainting with diffusion model. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 22428–22437, 2023. 2

[50] Ziyang Yan, Lei Li, Yihua Shao, Siyu Chen, Zongkai Wu, Jenq-Neng Hwang, Hao Zhao, and Fabio Remondino. 3dsceneeditor: Controllable 3d scene editing with gaussian splatting. *arXiv preprint arXiv:2412.01583*, 2024. 1

[51] Lihe Yang, Bingyi Kang, Zilong Huang, Xiaogang Xu, Jiashi Feng, and Hengshuang Zhao. Depth anything: Unleashing the power of large-scale unlabeled data. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 10371–10381, 2024. 2

[52] Yunhan Yang, Xiaoyang Wu, Tong He, Hengshuang Zhao, and Xihui Liu. Sam3d: Segment anything in 3d scenes. *arXiv preprint arXiv:2306.03908*, 2023. 2

[53] 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. 1, 2, 3, 4, 5, 6, 7

[54] Youtan Yin, Zhoujie Fu, Fan Yang, and Guosheng Lin. Or-nerf: Object removing from 3d scenes guided by multiview segmentation with neural radiance fields. *arXiv preprint arXiv:2305.10503*, 2023. 2

[55] Jiahui Yu, Zhe Lin, Jimei Yang, Xiaohui Shen, Xin Lu, and Thomas S Huang. Generative image inpainting with contextual attention. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 5505–5514, 2018. 2

[56] Guanhua Zhang, Jiabao Ji, Yang Zhang, Mo Yu, Tommi S Jaakkola, and Shiyu Chang. Towards coherent image inpainting using denoising diffusion implicit models. In *International Conference on Learning Representations (ICLR)*, 2023. 2

[57] Richard Zhang, Phillip Isola, Alexei A Efros, Eli Shechtman, and Oliver Wang. The unreasonable effectiveness of deep features as a perceptual metric. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 586–595, 2018. 6

[58] Siting Zhu, Guangming Wang, Hermann Blum, Jiuming Liu, Liang Song, Marc Pollefeys, and Hesheng Wang. Sni-slam: Semantic neural implicit slam. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 21167–21177, 2024. 2

[59] Zihan Zhu, Songyou Peng, Viktor Larsson, Weiwei Xu, Hujun Bao, Zhaopeng Cui, Martin R Oswald, and Marc Pollefeys. Nice-slam: Neural implicit scalable encoding for slam. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 12786–12796, 2022. 3

[60] Xueyan Zou, Jianwei Yang, Hao Zhang, Feng Li, Linjie Li, Jianfeng Wang, Lijuan Wang, Jianfeng Gao, and Yong Jae Lee. Segment everything everywhere all at once. *Advances in neural information processing systems*, 36: 19769–19782, 2023. 2